PROCEEDINGS

NOVEMBER 28-DECEMBER 1, 1994

16th

N S I A

PROCEEDINGS

16TH INTERSERVICE / INDUSTRY TRAINING SYSTEMS AND EDUCATION CONFERENCE

Accesion For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

November 28 - December 1, 1994

FORWARD

The *Proceedings* of the 16th Interservice/Industry Training Systems and Education Conference (I/ITSEC) contain all papers to be presented. The success of last year's poster displays led us to a specific session allocated to poster papers. This allows authors to provide an in-depth discussion of their research. This year's papers are presented in six tracks:

Policy and Management
Education, Instruction and Training Methodology
Training, Development and Delivery
Modeling and Simulation
Simulation and Training Systems
Research & Development Technology Applications

The Conference Committee listed on the following pages devote a great deal of time and effort to make this conference a success and they have my sincere appreciation. Each year we try to present innovative approaches and solutions to current problems. Please share your ideas for future conferences by completing the forms provided in each session.

On behalf of the entire committee, we hope you enjoy the conference.

Ed Ward
Program Chair

INTERSERVICE EXECUTIVE COMMITEE

Chair

Captain Christopher L. Addison, USN
Commanding Officer, Naval Air Warfare Center Training Systems Division
Orlando, Florida

Brigadier General John F. Michitsch, USA
Commander, US Army Simulation, Training and Instrumentation Command
Orlando, Florida

Colonel Robert R. Tyler, USMC
Liaison Officer/Program Director Marine Corps Systems
Naval Air Warfare Training Systems Division
Orlando, Florida

Dr. Robert R. Barthelemy
Product Group Manager, Training Systems Program Office
Air Force Aeronautical Systems Center
Wright-Patterson AFB, Ohio

Dr. Harold F. O'Neil, Jr.
Professor of Counseling and Educational Psychology
School of Education, University of Southern California

INTERSERVICE STEERING COMMITTEE

Chair

Carol Denton
Head, Instructional Systems Surface & Submarine Division
Naval Air Warfare Center Training Systems Division

Ralph C. Nelson
Chief Plans & Operations, US Army Simulation, Training and Instrumentation Command

Stephen Miller
Deputy Director, US Air Force Flight Training System Program Office

Major Lawrence "Rick" Wrenn, USMC
Project Manager, Naval Air Warfare Center Training Systems Division

Frank Schufletowski, Ph.D.
USAF Instructional Systems Development
Directorate of Plans & OperationsHQ Air Education and Training Command

Pam Bridges
Head of Contract Management Branch
Chief of Naval Education and Training

CONFERENCE COMMITTEE

Conference Chair
G. P. "Pres" McGee, Lockheed Fort Worth Co.

Program Chair
Ed Ward, Hughes Training, Inc.

Deputy Program Chair
Dennis Shockley, Evans & Sutherland

Subcommittee Chairs:

Conference Chaplain
Col. Alphonso "Al" Castellana, USMC (Ret.)

NSIA Representatives
Col. Bruce Green, USA (Ret.)

Romaine E. Gaskins
Asst. Director, National Activities

NTSA Representative
VADM John Disher, USN (Ret.)
Executive Director

Best Paper Subcommittee Chair
Dr. Robert Breaux

Facilities Committee
Ron Johnson

Publications Committee
Don Jacobs, PULAU Electronics Corporation
JoAnne Puglisi, Martin Marietta Information Systems

Scholarship Committee
Jack Drewett

Academic Advisor
Dr. A. Louis Medlin, Institute for Simulation and Training, University of Central Florida

Academic Liaison
Lt. Col. Milt Nielsen, Ph.D, US Air Force Academy

INSTRUCTIONAL TECHNOLOGY ADVISORY GROUP

Chair

John Buckley, USA, HQ TRADOC

Members

Gary Boycan
Office of the Secretary of Defense

Andrew "Andy" Andrews
Department of Energy, Los Alamos National Laboratory

John Hurlimann
American Academy of Distance Learning

Francis "Frank" McCoy
Naval Undersea Warfare Center

POLICY AND MANAGEMENT SUBCOMMITTEE

Chair
Len Kravitz, Reflectone, Inc.

Deputy Chair
Stan Aronberg, SIMTEC, Inc.

Members
Ann O'Kennon, EER Systems, Inc.
Bob Bret, CAE-Link Corporation
Bruce Schwanda, SIMTEC-Israel
Capt. John McGill, Chief of Naval Education and Training
Carla Knapp, Loral Federal Systems Co.
"Chip" Raymond Lanier, SAIC
Chuck Apfelbeck, Continental Dynamics, Inc.
Dave Manning, STRICOM
Dusty Woodlee, AAI Corporation
Ed Aufderheide, ARINC
Gene Holcomb, Harris Corporation
George "Pete" Hoyt, OCI
Gerry Balz, TRW-Orlando
James O"Bryant, STRICOM
Jay Erwin, C.O. Naval Air Systems Command
Jeff Marlin, Firearms Training Systems, Inc.
Johnie Herbert, Lockheed, FTW
LCDR Jeff Hesterman, Chief of Naval Operations
LTC Elaine Howell, USAF
LTC James "BUCK" Leahy, STRICOM
Maj. Pat Mullen, USAF
Ric Morrow, Hughes Training, Inc.
Richard Wagner, USAF
Sam Worrell, FAAC, Inc.
William L. Brigadier, TRW, Inc.

Section 1
Table of Contents
Policy and Management Papers

COMBINED TEST: A TEAM APPROACH TO ACHIEVING SIMULATOR-AIRCRAFT CONCURRENCY

Major William R. Corrigan
USAF
and
John K. Clapp
CAE-Link Corporation

ABSTRACT

Can Aircrew Training Device (ATD) testing be restructured to better support concurrent simulator-aircraft development and delivery to the using commands while reducing cost, mitigating schedule risk, and effectively using a reduced number of experienced test personnel? Traditional development and acceptance testing followed an iterative process of identical activities conducted first by the contractor then repeated by the Government. This inefficient process increased program cost and schedule risk. The reality of force downsizing has contributed to test risk by reducing the number of personnel available top support a traditional test program, especially a program seeking to achieve concurrency. To deal with these problems, the B-2 ATD Government-Contractor team developed a combined test methodology to eliminate redundant test, consolidate similar activities and complement the major program objective, concurrent development and delivery of the ATDs. The purpose of this paper is threefold: first, to identify the test related problems associated with concurrent development of complex training devices for a highly software-dependent aircraft not yet in flight test; second, to illustrate the team-oriented structure and process of combined test and how it proved critical to B-2 ATD delivery and functionality; and third, to present the results -- the on-time delivery of two B-2 Aircrew Training Devices that reflect the configuration and capabilities of the first operational B-2 delivered to Air Combat Command.

About the Authors

John K. Clapp is Test Manager for the B-2 Aircrew Training Devices at CAE-Link Corporation, Binghamton, New York. He served in the U.S. Air Force and retired as a Lieutenant Colonel. He has extensive experience in Aircrew Training and Aircrew Training Devices and has managed Air Force Operational Test for a number of ATDs including the B-52 Weapon System Trainer (WST), F-16 WST and B-1B WST.

Major William R. Corrigan is Deputy Subteam Leader and Test Director for the B-2 Aircrew Training Devices (ATDs) at the B-2 System Program Office, Aeronautical Systems Center, Wright-Patters Air Force Base, Ohio. In conjunction with operational and command-level assignments as a B-52G Electronic Warfare Officer and B-1B Defensive Systems Officer, Major Corrigan spent the last fourteen years as a Training Systems Acquisition and Test Manager for the B-52G, B-1B and B-2 ATDs. Other related positions included instructional systems designer, curriculum development manager and Initial Operational Test and Evaluation Director for the B-1B Weapon System and Cockpit Procedures Trainers. He received a B.B.A. from the University of Massachusetts and a M.S. from Binghamton University.

COMBINED TEST: A TEAM APPROACH TO ACHIEVING SIMULATOR-AIRCRAFT CONCURRENCY

Major William R. Corrigan
USAF
and
John K. Clapp
CAE-Link Corporation

THE CONCURRENCY CHALLENGE

From the very beginning, the B-2 Aircrew Training Device (ATD) program faced a major challenge -- to develop and deliver a concurrent ATD prior to deployment of the first operational aircraft. While the parallel procurement of a new weapon system and its associated training system is now an accepted practice, this process was largely untested when the B-2 ATD program began in 1985.

On paper and in a perfect world, parallel development and delivery of both aircraft and training devices is possible. Figure 1 depicts specific paths for each development program that could be taken to achieve concurrency, paths tailored to a program that uses actual aircraft Operational Flight Programs (OFPs) as did the B-2 ATD program. The design and development of the aircraft OFPs parallels the design and development of the supporting ATD hardware, software, instructional features, and environmental factors unique to the simulator are initiated.

Upon simultaneous release of the qualified OFP to flight test and the ATD, the code is tested and integrated into the WST to assure compatibility with ATD interfaces, environmental factors and simulated aircraft functionality. Upon conclusion of flight test, the same basic capability has been integrated, tested in the ATD and accepted by the Air Force. When the aircraft is in the Rework/Update stage, the ATD receives those updates prior to related aircraft flight test and integrates them, with appropriate regression testing, in line with aircraft flight test to assure the same functionality in the ATD as the delivered air vehicle.

In addition to these complex development activities encountered by the B-2 ATD program, the ATD test requirements were greater than those encountered in simulator programs for established aircraft. Not only were we required to test the normal functionality associated with high fidelity flight simulators (i.e., motion, aural cue, visuals, etc.), but we were faced with a brand new radar with very unique capabilities and the most complex OFPs yet fielded for any aircraft. The unique aerodynamic performance of the 'flying wing' also added additional test procedures to assure faithful representation of the aircraft's flight characteristics. All of this had to be accomplished under severe personnel constraints driven by the limited number of aircrew members qualified in the B-2 or even familiar with the B-2 systems.

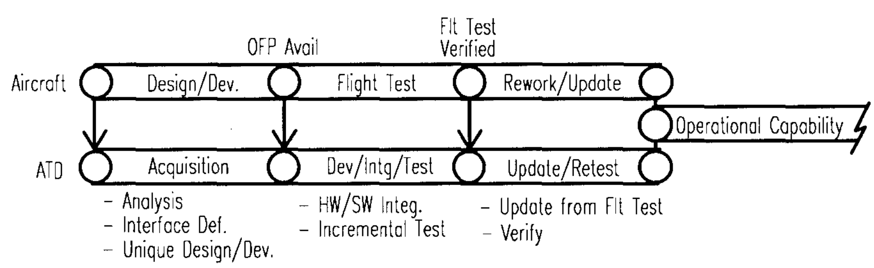


Figure 1
Concurrency

To successfully accomplish a concurrent development is a challenge that requires a relatively stable baseline -- and OFP stability was a challenge as the B-2 aircraft program drove toward initial aircraft delivery. There were three (3) major OFP releases before delivery of the first aircraft, releases involving significant software changes as a result of flight test and the need to accommodate maturing functionality. This incremental delivery of aircraft software was not part of the ATD plan and "broke" our schedule. Each new OFP update required time to integrate into the ATD, and in some cases required redesign of the ATD interface software. All updates required some degree of regression testing activities.

It was obvious that something significant had to be done if we were to incorporate the aircraft changes and still deliver an ATD 90 days prior to the arrival of the first aircraft. The incremental OFP delivery schedule caused our original test and delivery plan, which assumed only one OFP delivery, to be unable to meet program requirements. To deliver and meet the user's needs dictated that time savings be made in the test program and its schedule.

THE PROBLEM - A TRADITIONAL TEST PROGRAM

The original B-2 ATD test program was structured in line with a traditional process required by the Statement of Work (SOW). As illustrated by Figure 2, the test program consisted of a succession of serial and often redundant test activities designed to ensure that the delivered product met the requirements of the system specification as well as the training needs of the user.

The original plan required extensive contractor engineering activities to integrate and verify system performance (including running of the test procedures by the individual engineers). The formal test program then began with Contractor In-Plant Verification Testing, which was the complete accomplishment of the proposed Development Test Procedures (DTPs) by Link Quality Assurance (QA) personnel and witnessed by designated Government quality representatives.

After completion of Verification Testing and correction of all deficiencies identified during this phase, Government Development Test and Evaluation (DT&E) began with a Computer Program System Generation (CPSG). The CPSG or Cold Start was followed by Government In-Plant Development Performance Testing, which was the Air Force accomplishment of selected systems and subsystem tests contained in the DTP. The historical reasons for this form of DT&E are valid, but there is no denying the redundancy of this test activity in light of the previously completed Verification Testing. Once all specification testing was complete (Verification and DT&E), in-plant Initial Operational Test & Evaluation (IOT&E) was planned to evaluate the ATD's operational effectiveness and suitability, as well as to ensure that the first delivered devices met the user's requirements.

The same process of Verification Test, DT&E and IOT&E was repeated on site. Early program schedules included up to 25 weeks of formal in-plant test activity. In addition, the on-site test requirements helped stretch the teardown, pack, install, checkout and acceptance of the ATD to 22 weeks. In all, approximately 11 months from start of in-plant test to final acceptance for each device.

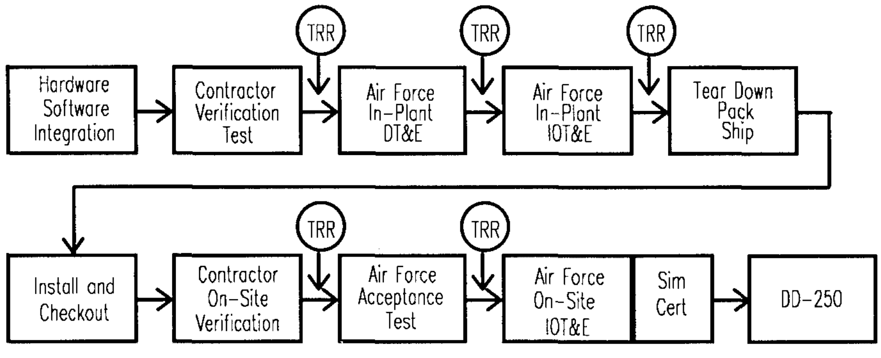


Figure 2
Traditional Test Approach

But time was not the only problem with our original test approach. Each of these serial, and often redundant, test periods was preceded by a readiness review to address a specified set of extensive and often confusing criteria that had to be met in order to continue. Because a different agency was responsible for each phase of the formal test (Contractor QA and "designated government representative" for Verification Testing, SPO Engineering for DT&E and AFOTEC for IOT&E), there were different "agendas" and large variances in methodology for each respective test period. While not a given, the existence of separate tests by separate agencies with separate goals can be extremely inefficient, and not necessarily conducive to success.

As delivery of the first B-2 to Whiteman AFB approached, and the B-2 ATD program moved toward initial training device delivery, it became apparent that the traditional test flow needed a complete overhaul to meet the challenge of an accelerated, dynamic and fiscally constrained program. Test conduct had to be streamlined, test time had to be reduced, all while test quality remained at the highest level. To deliver a quality product under the constraints of the program required a new way of doing business.

THE TEST SOLUTION - TEAMWORK

In late 1991, representatives of the Air Force and CAE-Link met to construct the new test plan and schedule while retaining the goals of a traditional test program through a more efficient approach. All parties recognized the need for innovation and cooperation and put aside individual organizational preferences, requirements and "turf" issues. Discussion ultimately led to a key agreement that all test agencies had equal responsibility for the successful completion of test -- success defined as on-time delivery of a thoroughly tested ATD that met the user's requirements. This led to the formation of a quasi-formal committee charged with all test responsibilities -- the B-2 ATD Joint Test Management Team (JTMT) (See Figure 3). The JTMT was made up of Air Force and Link organizations charged with primary B-2 ATD system test responsibility. Under the JTMT concept:

a. CAE-Link Engineering is responsible for building and integrating the ATD, developing the test procedures, and correcting any deficiencies found during test. Engineering support of actual test conduct is also provided.

b. CAE-Link B-2 ATD Quality Assurance is responsible for accomplishing/witnessing conduct of Verification Testing and any other testing using the DTP.

c. CAE-Link B-2 Program Test Manager is responsible for coordination of all Link formal test activities and providing the Link single point of contact for test.

d. The B-2 System Program Office is responsible for Air Force management of the B-2 ATD Test Program. The SPO Test Manager is responsible for coordination of all Air Force activities and decisions during test.

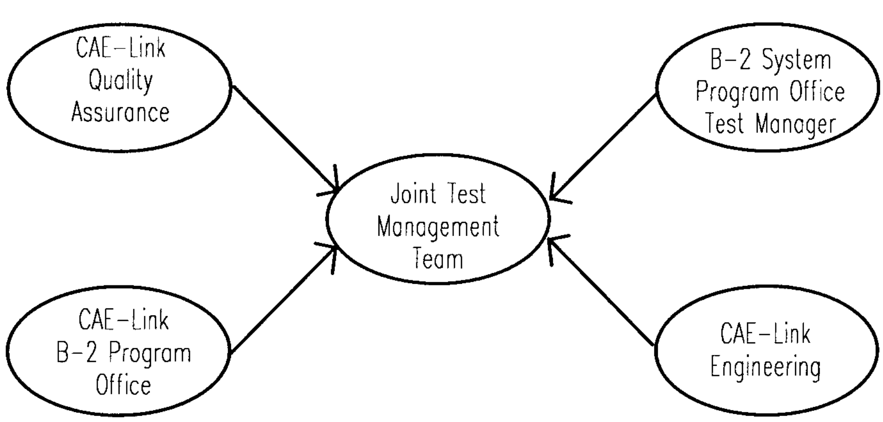


Figure 3
Joint Test Management Team (JTMT)

The members of the JTMT shared equally in decision making and the responsibility for test success. Unanimous decisions were the desired outcome but the Air Force Test Manager, and not other JTMT members, retained "Veto Power". In cases where agreement could not be reached, the Air Force was "most equal". Provisions were made for the Air Force OT&E Test Director to sit as part of the team during periods of IOT&E and to participate, as desired, during other periods of test.

Once formed, the JTMT set to work to restructure the test program, scheduled to begin in just four months. In its efforts, the team drew upon its significant collective experience in simulator development and test and the experience and work of others in this arena. One work, the process known as "Simulator Test 2000", developed by an Air Force - Industry Critical Process Team under the auspices of Total Quality Management, played a role in the restructure of efforts. The basic concept of "Simulator Test 2000", with its early assessments, elimination of redundant test activities, and high-level mission activities, was used as we developed the B-2 ATD Combined Test Program depicted in Figure 4.

One of the first requirements for an efficient yet complete test, the Development Test Procedures (DTP), had already been constructed. Written at a relatively high level, they assumed a basic level of system knowledge by the test participants. While this approach resulted in some Test Discrepancies (TDs) due to operator inexperience, the benefits gained in efficiency, as well as a more dynamic and realistic exercising of the system, more than compensated for the few extra TDs. The JTMT next reviewed Link's Configuration Management and Load Build process to reaffirm an earlier Test Planning Working Group decision that a CPSG was unnecessary and need not be accomplished.

The serial and redundant nature of Verification testing and Government DT&E was the next issue we tackled. The solution was straightforward -- combine these two activities. However, the Air Force retained the right to conduct any additional tests it felt were necessary. During Combined Test, the DTP was run jointly by Link QA and designated Air Force representatives and system experts. Any TDs written carried two signatures and subsequent rechecks of these TDs carried two signatures. Side-by-side accomplishment of the DTP and the hands-on management of test by the JTMT allowed the B-2 SPO to grant credit for DT&E up front, saving significant time and effort.

Having decided on the process we termed "Combined Test", the JTMT still had to produce a detailed plan to accomplish the full DTP in a more efficient manner while retaining high confidence in the quality of test. This task was even more important since the team wanted to assure that the test was sufficient to preclude exercise of the Air Force's "additional test option". This was accomplished by first grouping the DTP sections into logical blocks and applying historical time factors based on engineering dry runs, and then developing a worst case test sequence. This worst case scenario produced a new serial combined test of 24 weeks duration, one week less than the traditional approach, but too long under our current schedule constraints.

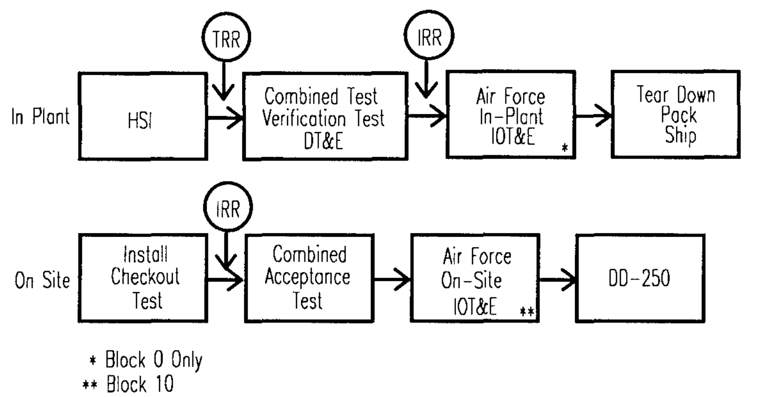


Figure 4
Combined Test Program

Additional "scrubbing" of the test procedures identified specific tests that could be run in parallel or simultaneously (e.g., standalone radar and aero performance tests which do not require use of the ATD). A number of tests that could be run more efficiently during teardown or installation on-site were deferred. Finally, results of engineering dry run times were applied to the remaining tests to provide better run time estimates. The resultant Combined Test period became 13 weeks (an 11 week reduction) for the initial functionality of the B-2 WST, a capability we call Block 0, which represented the first OFP to enter flight test.

Having optimized the development test activities, the more objective DT&E tests, the team then considered the more subjective but equally important Initial Operational Test and Evaluation activities, scheduled to be conducted for two weeks following Combined Test of the Block 0 WST. Considering the different objectives and methodology of IOT&E, the JTMT's goal was to exercise the system in a mission environment as early as possible to identify problems that might not be found while running a specific test procedure but might well be found during IOT&E. This was accomplished by conducting Link Mission Test in conjunction with Combined Test. At least once a week throughout the test period, the aircrew experienced personnel from Link's Operational Training Analysis Group (OTAG) flew a mission scenario approximating flight conditions expected during actual Air Force training. When discrepancies of any kind were identified, they were documented and processed just like any other Test Discrepancy. In addition, comments and observations made during Familiarization Training for the Air Force Test Team were processed as TDs to further identify, track and resolve operational type problems early in the test program. Finally, Air Force aircrews were the designated government representatives during the Integrated System DTP tests, providing additional early operational experience and knowledge. By adding an operational flavor into the test program in its early stages, the test team identified a significant number of operational issues and corrected them before the actual IOT&E, thus saving time and providing a device much more representative of the B-2 at the end of the initial test period. The Air Force conducted IOT&E on the Block 0 WST for seven days and wrote 80 Test Discrepancies, certainly far less than would have been written had this operational flavor not been inserted into the test at an early date.

In all, the B-2 WST Block 0 Combined Test Team accomplished almost 200 separate test procedures plus Initial Operational Test and Evaluation in 16 weeks, a savings of 8 weeks over the original plan and schedule. Running the DTP produced 501 Test Discrepancies and an additional 268 were generated during Link Mission Test, Familiarization Training and IOT&E. This number represents a significant reduction in the number of TDs written in the past on devices of similar complexity.

The B-2 Program's initial experience with Combined Test was not without problems, two of which detracted from the overall success of the effort. The first stemmed from our failure to develop a process/procedure to define the roles and responsibilities of all test participants, particularly the role of the JTMT. This led to protracted discussions involving Test Discrepancy resolution and the conduct of the test, more from a "turf" perspective than a results issue. Prior to the subsequent round of testing, a procedure was written to satisfy all agencies. This procedure allowed each to meet its goals but established the JTMT as the final authority in all test matters.

The second failure was evident by a high rejection rate during TD rechecks, a rate that rose above 25%. We corrected this by adding an additional review of the TD before it was declared Ready for Recheck (RFR), scheduling rechecks in logical blocks of related TDs, and ensuring engineering support during the recheck. By tightening control of the recheck process, the JTMT spent a little extra time up front which saved time at the end and resulted in a Block 10 rejection rate of 6%.

With Block 0 experience and lessons learned, the JTMT turned to the real challenge -- to thoroughly test, both in plant and on site, the first delivered WST and MT. This test was of a new OFP configuration called Block 10, and was done under severe schedule constraints driven by delivery of the first aircraft to Whiteman AFB. The Block 10 configuration was a complete update to the previously tested system, involving new tests as well as regression testing in all functional areas. In addition, the ATD's radar system was being tested for the first time as was the first WST delivered to Whiteman AFB.

Using the same groundrules developed Block 0 test, the JTMT developed a program that tested a limited number of hardware-specific areas prior to and during teardown, and incorporated the remainder into the installation and checkout process. The largest test efforts, the functional aircraft systems, avionics and radar tests were laid out as blocks of related tests in a series of waterfalls that established at which point a function tested in plant would be shipped to site for verification on those devices. Because the on-site devices would be used by Northrop to validate the total training course, our requirement was to have them fully tested in a Block 10 configuration and verified using Air Force crews. Through judicious application of the concepts of combining test, flexible scheduling, juggling of people and plain hard work, our ambitious goals were actually exceeded. DT&E and IOT&E were accomplished on the delivered devices ahead of schedule with significantly fewer TDs than anticipated and the feedback from the using command was extremely positive.

THE RESULTS

It is difficult to compare the earlier B-2 ATD program plan (pre "Block 0") because of different requirements, strategies and processes, as well as different aircraft development plans, that impacted the ATD program. Nevertheless, analyzing the earlier program, we determined that at least 90 weeks were scheduled to test the two delivered devices in plant and on site through two different OFP configurations. By combining test activities, testing off-line where appropriate, assigning procedures to a device and allocation, obtaining early operational input, and constantly working to optimize the test program, the B-2 ATD Test Team was able to deliver two devices ahead of schedule. The total time required for all Block 0 and Block 10 testing, both in plant and on site, was 55 weeks, a reduction in test time of approximately 39%. The nearly nine (9) months saved, permitted the B-2 ATD program to recover and "mend" its schedule that was broken due to aircraft program accelerations and additional OFP deliveries. The dollar savings to the ATD program through the restructure of the test program was initially estimated at $7.5 million.

Saving schedule and money is a significant accomplishment but has no value if the product suffers. The B-2 ATD did not suffer from the efficiencies achieved during Combined Test. Not one test procedure was eliminated -- each procedure was run and every point in the test matrix was accomplished. The overall quality of the product in test was proven by the number and the priority of the Test Discrepancies written and the successful running of the Test Procedures. Almost 11,000 pages of DTP were run during the entire test and 1,235 TDs were written, a ratio of .11 per page, far below the 1.1 or higher experienced on other high-complexity programs. Of these discrepancies, approximately 20% were correctable with a documentation update and not a software or hardware change to the system. During three separate periods of Air Force IOT&E, crewmembers wrote 185 TDs. This number is far below the norm, due in part to the additional 188 TDs identified during Link Mission Test and Familiarization Training. The insertion of an early operational flavor paid big dividends for this program. One final but significant fact is that not one Priority 1 Test Discrepancy that would have stopped test was written during the 55 weeks of formal B-2 ATD testing.

Impressive as these numbers are, the important question is "Did the test support delivery of a concurrent ATD?". The answer is an unqualified Yes. On 20 September 1993, the WST, concurrent with delivered aircraft configuration, was turned over to the 509th Bomb Wing ninety (90) days prior to arrival of the first aircraft. Although only partially tested, this WST had been flown by Air Force crews and determined to be representative of the aircraft. By 17 December, the current Block 10 configuration was completely tested on a WST in-plant and that software sent to Whiteman AFB where it was again determined to replicate the aircraft. In January 1994, the 509th began using the WST and MT with fully tested and concurrent software. By the time the first class of students began to use the delivered WST and MT, a full test of these devices had been completed and both matched the configuration of the operational aircraft.

SUMMARY

Combined Test was a success for the B-2 ATD program for one reason -- teamwork. Each participant was willing to take responsibility for its success or failure. The Joint Test Management Team functioned smoothly in an environment of open communication and trust. In the midst of a tight schedule, problems had to be resolved quickly; time could not be wasted with parochial disagreements. The JTMT attacked problems head on to reach decisions and the Air Force never did use its "veto power". This and other benefits of the Combined Test approach continue as this is written. An update to the Block 10 aircraft OFPs is being incorporated into the third delivered aircraft. This update is implemented in the in-plant WST and will be tested starting late June 1994. We intend this test to follow the philosophy now standard for the B-2 ATD Combined Test to further streamline test time by reducing on-site test activities to a minimum number of DTPs and relying on Operational Test to verify ATD concurrency with the aircraft.

While Combined Test was a success for the B-2 ATDs, it is also applicable to the vast majority of training system acquisition programs and should provide similar rewards. Program Managers and their Test Directors and the contractors must consider adopting this method of test -- certainly customization for particular programs may be necessary, but the framework presented here provides the basis for implementation. The payoffs are significant and it may mean the difference between a satisfied customer or an improperly prepared student, of a Ready for Training device as opposed to an Available for Training device.

TRAINING SYSTEMS VIA "NEW WAY" BEST VALUE CONTRACTING AND MIL-STD-1379D

Steven L. Griffin
William Kitterman

Naval Air Warfare Center Training Systems Division
12350 Research Parkway
Orlando, Florida 32826-3224
407-380-8519

Neal M. Finkelstein
Simulation, Training, and Instrumentation Command
Orlando, Florida 32826-3224

ABSTRACT

In 1993-1994, STRICOM formed a team to acquire the AGTS (Advanced Gunnery Training System). The team developed a "new way of doing business" which synthesized a number of concepts--best value source selection; emphasis on processes and metrics and total quality leadership; concurrent engineering; integration of MIL-STD-1379D and the systems approach to training; application of the Fixed-Price-Incentive (Successive Targets) contract type; range pricing; and a uniquely structured Request for Proposal. This "new way of doing business is described and lessons learned are presented.

ABOUT THE AUTHORS

Steven L. Griffin is an Instructional Systems Specialist, NAWCTSD. He holds an M.Ed. from the University of Louisville. Mr. Griffin has twenty-four years of Government and industry experience in concept formulation and training system and courseware development.

Mr. William Kitterman is a Contract Specialist and limited ($10M) Contracting Officer for the Close Combat Training System Branch, Land Warfare Contracts Division, NAWCTSD. He holds an M.B.A. in Procurement and Contracting from George Washington University. Mr. Kitterman has twenty-five years experience in Government and industry. Mr. Kitterman was the Contract Specialist throughout the Advanced Gunnery Training System source selection process.

Neal M. Finkelstein is AGTS Project Director for the Close Combat Training System Division, STRICOM. A doctoral student in Industrial Engineering at the University of Central Florida, he holds an M.S. in Industrial Engineering from Texas A&M University, and a B.S.E.E. from Florida Atlantic University. Mr. Finkelstein was formerly a Program Manager for the Hybrid Microelectronic PATRIOT Fuze Program at Harry Diamond Laboratories.

TRAINING SYSTEMS VIA "NEW WAY" BEST VALUE CONTRACTING AND MIL-STD-1379D

Steven L. Griffin
William Kitterman

Naval Air Warfare Center Training Systems Division
Orlando, Florida

Neal M. Finkelstein
Simulation, Training, and Instrumentation Command
Orlando, Florida

INTRODUCTION

In simplified terms, acquisition of a training system may be viewed as having three phases: (1) Development and release of a Request for Proposal (RFP) built from the user's requirements, (2) Proposal evaluation and contract award, and (3) Government and awardee accomplishment of the development, testing, fielding, and life cycle support of the system. The Advanced Gunnery Training System (AGTS) program has completed phases one and two, with some significant accomplishments and lessons learned.

BACKGROUND

The AGTS program will provide training systems to support individual, crew, section, and platoon gunnery training for Army personnel who operate the M1A1 and M1A2 Abrams tank, the M2/M3A3 Bradley vehicles, and the Armored Gun System.

THE "NEW WAY OF DOING BUSINESS"

As the scriptures tell us, there is really nothing new under the sun. The AGTS approach synthesizes many concepts which may be familiar: best value source selection, processes and metrics and continuous improvement; concurrent engineering (CE); integration of MIL-STD-1379D and the systems approach to training (SAT); application of the Fixed-Price-Incentive (Successive Targets) (FPIS) contract type; range pricing; and a uniquely structured RFP package. It was the synthesis of these elements which resulted in a new paradigm and a new attitude.

"BEST VALUE" SOURCE SELECTION

Creech (1992), noting that best value source selection has "been around since the forties," describes it as follows: "A process based upon the use of reasoned judgement in selecting for contract award that firm whose proposal reflects the optimum combination of functions, features, performance, and price...." (p. 2).

Glennon and Fagan (1984) pointed out a natural affinity between the best value approach and the front end analysis required to support the development of a training system:

> The Best Value Acquisition Strategy includes the goal to specify at a relatively high level of abstraction. Therefore, once the decision is made to avoid specification in equipment terms (how), the specification writer has no choice but to concentrate intently on the results of Front-End Analysis. In other words, the Best Value Strategy provides a forcing function for a thorough Front-End Analysis and a clear definition of the training problem and associated constraints. (p. 182)

Certainly, the AGTS team found this statement to be correct. The AGTS approach, however, went further: Analysis was viewed as a continuing, not just a "front end" requirement, due to the evolving nature of the vehicles and tasks involved, and a need to continuously insert improving technology, including instructional technology. Additionally, best value was seen as providing an impetus to continuous incorporation of all phases of the SAT, not just the analysis phase.

PROCESSES AND METRICS

One clear link between the AGTS RFP and the literature of total quality leadership (TQL) is the importance of the concepts of **process** and **metrics**.

These two concepts appear throughout the AGTS RFP. For example, the instructions related to the SEMS (Systems Engineering Master Schedule) stated "The SEMS shall at a minimum consist of the offeror's tasks, major subcontractors' tasks, milestones and criteria (measurable metrics) for acceptable accomplishment at each milestone." Similarly, the evaluation factors for award included the following: "Each proposal will be evaluated from the standpoint of adherence to sound practices...(and the) extent to which the offeror has developed measurements to track the process, eliminate errors, remove slack, reduce variation, and plan for continuous improvement."

It was the desire of the AGTS team to achieve the effect summarized by Commander John Langford:

> Processes, measures, customer focus, teaming, and empowerment are all necessary for laying a foundation for continuous improvement...The criteria for empowering a team is fairly straightforward: The team must fully understand customer requirements; understand and accept their responsibilities, authorities, and accountabilities; have the processes, metrics and requisite skills necessary to perform their tasks; and have the full support of upper management. (1994, p. 1)

CONCURRENT ENGINEERING (CE)

Another concept which appears as a requirement in the RFP is CE. CE emerged as a named concept in 1988, as a result of efforts by industry and Government personnel to overcome two weaknesses in traditional systems engineering: the sequential, linear nature of the process; and the proliferation and isolation of specialists. Linton (1991) explains CE in the following terms: "CE involves a product development infrastructure that fosters a unified, collaborative approach that integrates inputs from business, engineering and management specialists across the traditionally segregated phases of product development (p. iv)."

MIL-STD-1379D

Just as there is a close affinity between the "new way of doing business" and TQL, so there can be a mutually supportive relationship between both of these and a careful application of MIL-STD-1379D, Military Training Programs.

In the AGTS RFP, the Government included two products from MIL-STD-1379D as mandatory requirements: the Training Situation Analysis and the Training System Alternatives Report. Otherwise, the RFP allowed each offeror great freedom:

> Describe the strategy for applying the SAT and MIL-STD-1379D in the analysis, design, development, implementation, and evaluation of courseware for AGTS... The offerors shall provide a Data Item Description for each proposed item of courseware...Offerors are invited to propose substitutions or exceptions to the Work Statement and contract data requirements list (CDRLs).

In response to this empowerment, each offeror, in a variety of approaches, performed some or all of the following tasks: analyzed anticipated AGTS program requirements; selected and tailored tasks and DIDs from MIL-STD-1379D; prepared CDRLs; integrated the tasks and data products into the SEMS and Systems Engineering Master Plan (SEMP); provided additional process information, if needed; and provided appropriate metrics. The results (the submitted proposals) were much more satisfactory than the results of many a Government "CDRL-scrub."

SYSTEMS APPROACH TO TRAINING (SAT)

For the Army, the SAT is defined primarily in TRADOC Regulation 350-7. The model presented has five interrelated, nonlinear phases: analysis, design, development, implementation, and evaluation.

The SAT links well to the "new way of doing business," for several reasons: It integrates systems engineering approaches into training system development, and provides the framework within which MIL-STD-1379D can be applied. Its iterative nature provides for continuous reexamination and improvement of the training system development. Finally, the regulation itself is based on baseline processes and metrics: for each phase, key processes and minimum essential requirements (MERS) are identified.

BUSINESS AND CONTRACTUAL ASPECTS

The selection of the AGTS contract type arose from a desire to synthesize the ideas above into a "new way of doing business." The train of thought went something like this:

* First was the desire to use the SAT as the core concept of contract performance.

* CE would be required to implement the concept.

* For CE to work, the contractor had to be pursuing **his** solution that he believed in. The Government had to cease dictating solutions and aproaches.

* The contract had to support and satisfy multiple users with diverse needs over a lengthy period of time.

* Access to technology insertions and business flexibility was needed.

* The use of other than Firm-Fixed-Price together with the desire that the contractor be executing his concept, led to the "best value" source selection method.

All the requirements had to be achieved under competitive conditions, leading to a "program friendly" production contract that was an attractive business arrangement for industry. Where development is not pushing the state of the art, the use of a production contract form lends program stability (through budgeting certainty), and allows for a lengthy contract, giving the contractor some assurance of future business base.

The need for flexibility and equity led to a selection of range pricing and the little used FPIS contract type.

FPIS is a standard Federal Acquisition Regulation contract type (FAR 52.216-17). Fixed Price Incentive (FPI) contracts are characterized by three things: (1) A ceiling price that is not subject to change, hence "fixed price"; (2) A target price that is the anticipated price of performance; (3) An overrun/underrun cost share ratio (which may be different for overruns and underruns), hence the "incentive."

All FPI contracts provide for the negotiation of final firm fixed prices. The successive target type is distinguished from the fixed target type by provision for an interim negotiation at a specified milestone which will convert the contract to either a Fixed Price Incentive (Firm Target), [the successive target] type; or a Firm Fixed Price type, with the latter preferred. The FPIS type also typically has a higher ceiling and a larger Government share than the FPIF type. Both types are designed for an equitable division of risk between the contractor and the Government.

In summary, the FPIS contract type allows for the equitable sharing of cost risks early in the contract when there are many unknowns, and provides for negotiation of fixed prices reasonably early in the contract after the principal risks are known.

Range pricing is simple in concept. The contractor proposes different unit prices for different ranges of option quantities. In AGTS the range pricing table included in Section B of the RFP was quite complex because it allowed for ordering almost any mix of <u>twelve</u> different configurations delivered to any of ten destinations, with unit prices for quantity ranges of 1, 2-4, 5-8, and 9-24. This provided the Government with immense flexibility in exercising options.

RFP STRUCTURE

Like contract type, RFP structure was selected to reflect the "new way of doing business." The unique aspects of key sections of the AGTS RFP are discussed in the following paragraphs.

Section L

Sections L was written to require the offerors to build their proposals as follows: Volume I, Past Performance; Volume II, Requirements Evolution; Volume III, Integrated Management; Volume IV, Supportability; Volume V, Affordability;and Volume VI, Administrative.

The use of a Past Performance volume reflected a best value source selection concept. This volume was analyzed by a separate evaluation group, using data provided by each offeror on Government contracts worked as a prime or subcontractor during the past three years.

Volume II, Section L, Requirements Evolution, required the offerors to fully lay out their pre-contract and post-contract approaches for integrating systems engineering processes with the SAT and MIL-STD-1379D:

> Show how the results of the integrated systems engineering/training system requirements analysis process will lead to major system/subsystem functional requirements and to the overall training system requirements.
>
> Discuss the strategy for applying the systems approach to training and MIL-STD-1379D in the analysis, design, development, implementation, and evaluation of courseware for AGTS.
>
> The offerors shall provide a Data Item Description for each proposed item of courseware as an annex to Volume II (exempt from page limitation).
>
> Explain the analysis and design methods which will be used to translate training requirements to performance and finally to a visual system design.

Volume III, Section L, Integrated Management, placed two primary requirements on the offerors, in an effort to seek implementation of the "new way of doing business." These two requirements were (1) SEMP; (2) SEMS.

The SEMP was to synthesize what would normally be separate specialty plans into a coordinated master plan for the integration of all program efforts. The SEMP was to describe, via text and diagrams, proposed processes and procedures to accomplish AGTS.

The SEMS was to consist of the offeror's tasks, major subcontractor's tasks, milestones, and **criteria (measurable metrics)** for acceptable accomplishment at each milestone.

Section M

Section M was written to clearly reflect best value source selection concepts. For example, the relative importance of the areas was as follows: the Requirements Evolution volume (which contained most of the material on the SAT and MIL-STD-1379D), along with the Integrated Management volume (which contained the SEMP and SEMS, with their emphasis on processes and metrics) were rated as most important. Affordability, along with Supportabilty, was rated as less important than the two top areas, and more important than Past Performance.

Work Statement

For AGTS a very minimal (20 page) Work Statement was developed. The offerors were given full freedom to propose changes to the work statement and submit them as part of the Administrative volume. During the pre-proposal briefing it was made clear to the offerors that MIL-STD-1379D did not contain all necessary processes and metrics required to develop AGTS and that therefore the offerors would have to take action to insert necessary processes and metrics into their proposals.

The following statements were used in the AGTS Work Statement, in regards to SAT and 1379D:

> The contractor shall perform the trade off studies necessary to determine the training strategy and then finalize the system design.
>
> The contractor shall conduct a training situation analysis of the types and levels of training.
>
> IAW Task 206, MIL-STD-1379D, the contractor shall identify the various elements, such as alternative features, capabilities, and characteristics, of the AGTS training system and analyze the effectiveness in meeting the training requirements (DI-ILSS-81086).

Specification Guide

The AGTS RFP did not contain a specification. Instead, each offeror was required to deliver a "starting point" specification as part of the proposal. A brief (11 page) specification development guide was provided. The Guide stated "To the extent that it is known, provide information that defines the proposed system and quantifies the performance level proposed...." During the face-to-face discussions, each offeror was cautioned not to build a specification which was prematurely detailed; any design decisions described had to be supported with training requirements analysis and trade study data.

System Requirements Document (SRD)

An SRD was developed by the AGTS team, based on meetings with the users and on the user-developed requirements documents.

Requirements statements in the SRD were limited to the top level of detail; for example: "The system shall provide the capabilities to monitor and evaluate the individual's, crew's, section's, and/or platoon's duties in response to fire commands, in a realtime manner." Thus the SRD allowed for contractor innovation, technical and instructional effectiveness breakthroughs, full implementation of the SAT, meaningful trade off analyses conducted through concurrent engineering and participation with the user, and an evolutionary approach to AGTS design.

EVALUATING PROCESSES AND METRICS, USING MIL-STD-1379D AND THE SYSTEMS APPROACH TO TRAINING

In order to evaluate each offeror's application of MIL-STD-1379D and the SAT, the instructional systems specialists on the AGTS team applied analysis of AGTS requirements, along with the DID selection and tailoring guidance in Appendix A of MIL-STD-1379D and the minimum essential requirements in TRADOC Regulation 350-7. The results were identification of two sets of "core" tasks and DIDs from MIL-STD-1379D. One set related to the application of SAT and MIL-STD-1379D in support of the development of the AGTS simulation system hardware and software components, e.g. the visual system, computational system, crew stations, instructor-operator station, and exercise generation system. A second set related to support of AGTS courseware, e.g. the gunnery exercises in the instructional subsystem, and the courseware for training of the instructor-operators and the scenario-generation personnel.

A similar "starting point" Government analysis was made to identify critical processes and metrics for the core DIDs. An example is shown in the following excerpt (metrics are in **bold)**:

1. TRAINING ANALYSIS PROCESS:

a. Training Situation Analysis (TSA):

(1) The TSA **process** must be **complete**: MIL-STD-1379D subtasks 101.2.1, 101.2.2, 101.2.3 required.

(2) **Completeness** of the TSA **product**, is required, according to the tailoring requirements on the CDRL.

(3) TSA **scope** and **iterative schedule** (as shown on CDRL and SEMS) must reflect the variety and changing nature of the training situations.

(4) **User coordination.**

(5) **Multidimensionality**, that is, alternative problem solutions must be compared on all critical dimensions: training effectiveness, cost, schedule risk, engineering, risk, MANPRINT, supportability, maintainability, reliability, and others as required.

Similar analyses were made to examine the tailoring of the actual DIDs from MIL-STD-1379D. None of these Government analyses were considered to be THE answer, but a starting point for discussion.

LESSONS LEARNED

Create Operational Definitions

Process and metrics were two

central concepts of the AGTS RFP. Communicating precisely the meaning of these concepts, however, proved to be a task of great difficulty. During AGTS RFP development and proposal preparation and evaluation the Government and contractor AGTS teams repeatedly grappled with trying to achieve and communicate a common understanding of these two terms. For example, in an attempt to be more clear, other terms were used to amplify the meaning of "process," words like "approach," "methods," "strategy," and "practices." In retrospect, however, these amplifications probably just increased the communication problem.

Dr. Deming was certainly familiar with this problem, and proposed a solution, the operational definition:

> Meaning starts with the concept, which is in somebody's mind, and only there: it is ineffable ....An operational definition puts communicable meaning into a concept...An operational definition is one that people can do business with (Deming, 1986, pp. 276- 277).

One lesson learned, therefore, was to include in future RFPs operational definitions of key concepts, and to seek meaningful discussions of these concepts as early as possible.

Improve Preproposal Conference

For the AGTS program, discussions at the pre-proposal conference were superficial. In contrast, the face-to-face discussions of clarifications and deficiencies were extraordinarily open and useful for both the Government and the offerors. A way must be found to have this meaningful and open discussion at the pre-proposal conference, or during RFP development, or at some other early point in the process.

Early discussion should contain expanded and clear presentations by each Government functional representative concerning what they will be looking for, perhaps on a factor by factor level. The focus should be on "sound practices" and appropriate metrics, not on design solutions for the particular program. The discussions should include a communications check, whereby industry presents their understanding of what was said, with follow-on Government-industry clarifications. This type of communication of expectations would reduce the hours devoted to writing and resolving clarifications and deficiencies, which were a large part of the cost of AGTS proposal evaluation. Needless to say it would also reduce the rework of the bidders.

Clarify Requirement for Metrics

The Government should clarify that metrics include much more than a software tool, or a list of products and dates for accomplishment. Metrics also include measurements of quality, communication, coordination, and impact/traceability. For example, it does not matter if a visual fidelity analysis is done on time if it is of poor quality or if it is not communicated in a timely manner to the various functional areas within the offeror's team, or if once communicated it is not used.

Emphasize Plans, Processes, Metrics

A clear requirement and methodology must be developed to ensure that the offeror's proposed processes and metrics are placed in the contract in a clear and consistent manner. The original intention for AGTS, with its brief Work Statement, was that the offerors would have two avenues for inserting their desired processes and metrics: first, inclusion in the SEMS and SEMP, which were to become part of the contract; second, as changes to the Work Statement. Neither of these approaches worked satisfactorily. The offerors placed their processes and metrics primarily in Volume II, Requirements Evolution, and Volume III, Integrated Management. As a result, the Government decided to incorporate the entire proposal into the contract. This is a less than ideal solution, due to the fact that it is inevitable that there will be inconsistencies between the various parts of the proposal. Lesson learned: at the pre-proposal conference, place much more emphasis on the importance and the role of the SEMS and the SEMP. Another approach might be to eliminate the Work Statement from the RFP and have the offerors submit a Work Statement of their own which must match the approaches described in the technical and management volumes, and must be consistent with the SEMS and SEMP.

SUMMARY

For the AGTS program a "new way of doing business" was used by the RFP preparation team and was embedded into the RFP and the proposal evaluation process. This new approach synthesized elements of a variety of concepts, approaches, and tools: best value contracting, processes and metrics, continuous improvement, MIL-STD-1379D, the systems approach to training, concurrent engineering, Fixed Price Incentive (Successive Targets) contract approach with range pricing, and an innovative RFP structure. The results of this new approach were thoughtful, innovative, affordable proposals designed to meet the Army's evolving needs for advanced gunnery training now and in the future.

REFERENCES

Creech, D. (1992, August 17). Best Value Bulletin No. 1.

Deming, W. Edwards. (1986). Out of the crisis. Cambridge, MA: MIT Center for Advanced Engineering Study.

Glennon, R. & Fagan, C. (1984). Toward the Improvement of Training System Acquisition: The Compatibility of the Best Value Acquisition Strategy with Front-End Analysis. Proceedings of the Interservice/Industry Training Equipment Conference and Exhibition, 6, 177-184.

Langford, J. (1994, March). Empowerment: tell them what you want, not how to do it. Naval Aviation Systems Team Forum, 3(3).

Linton, L., Hall, D., Hutchison, K., Hoffman, D., Evanczuk, S., & Sullivan, P. (1991). First principles of concurrent engineering. Washington, DC: CALS/CE Electronic Systems Task Group.

MOVING IN A NEW DIRECTION: TRAINING & SIMULATION TECHNOLOGY CONSORTIUM

Janet Weisenford
Naval Air Warfare Center Training Systems Division
Orlando, Florida

William F. Jorgensen
Training Simulation and Technology Consortium
Orlando, Florida

Abstract

The Training and Simulation Technology Consortium (TSTC) is a new model for transferring defense training and simulation technology involving a partnership between the federal government, industry, and a university. Members include three government agencies, four DoD based industries and a major university. These members determined that technology transfer would not occur without commercialization. This involves identifying new customers, understanding customer requirements, matching requirements to defense-based capabilities, and then developing the distribution and sales process. TSTC was established to support this commercialization process through the Advanced Research Projects Agency (ARPA) under the Technology Reinvestment Program (TRP).

This paper reports on the process of forming the consortium, the barriers surmounted, the results of the consortium's efforts and what the future holds for such efforts.

About the Authors

Janet Weisenford is the Project Manager for Technology Transfer at the Naval Air Warfare Center Training Systems Division. Before joining NAWCTSD, she held positions in the Office of the Secretary of Defense and the Secretary of the Navy's Office of Program Appraisal. She has a Masters of Public and International Affairs from the University of Pittsburgh and a Bachelor of Arts in Public Policy from New College, Florida.

William F. Jorgensen is the Director of Operations and Technology for the Training and Simulation Technology Consortium (TSTC) in Orlando, Florida. He has over fifteen years developing training and simulation solutions to meet DoD, industry, and educational program requirements. Mr. Jorgensen has an MA in Instructional Development and Technology from Michigan State University, a BA in Math from the University of Missouri and a BS in Science Education from Ferris State University.

MOVING IN A NEW DIRECTION: TRAINING & SIMULATION TECHNOLOGY CONSORTIUM

Janet Weisenford
Naval Air Warfare Center Training Systems Division
Orlando, Florida

William F. Jorgensen
Training Simulation and Technology Consortium
Orlando, Florida

The Training and Simulation Technology Consortium (TSTC) is a new model for transferring DoD training and simulation technology involving a partnership between the federal government, industry, and a university. Members include: the Naval Air Warfare Center Training Systems Division, the Army Simulation, Training, and Instrumentation Command, National Aeronautical and Space Administration, Loral Federal Systems, Analysis and Technology Incorporated, Dual Inc., Dynamics Research Corporation, and the University of Central Florida's Institute for Simulation and Training.

The TSTC members recognized that technology transfer would not occur without commercialization. New customers and markets needed to be located and new relationships had to be formed if defense training and simulation technology was to be applied in civilian sectors, both public and private. This concept was the basis for the TSTC proposal and was submitted to the Advanced Research Projects Agency (ARPA) under the Technology Reinvestment Program. This proposal was selected as one of the funded TRP projects.

This paper reports on the process of forming the consortium, the barriers surmounted, the results of the consortium's efforts and what the future holds for such efforts. The paper will also discuss the commercialization processes developed to date, the services available to the training and simulation industry from the TSTC, major milestones and achievements, and what the future holds for such efforts.

THE NEED FOR A TRAINING AND SIMULATION TECHNOLOGY CONSORTIUM

National Mandate

On February 22, 1993, President Clinton and Vice President Gore unveiled "Technology for America's Economic Growth, A New Direction To Build Economic Strength." This plan calls for:

-strengthening America's industrial competitiveness and creating jobs;

- forging a closer working partnership among industry, federal and state governments, workers, and universities;

- redirecting the focus of our national efforts toward technologies crucial to today's businesses and a growing economy;

- improving the skills offered by American workers by increasing the accessibility of education and training, and

- improving technology for education and training by supporting developments that increase the productivity of learning in schools, a variety of business training facilities and in homes (Clinton, 1993).

The Training and Simulation Technology Consortium, Inc. supports each of the five objectives cited in the President's plan. One of the keys to strengthening America's competitiveness lies in improving human performance. Improvements in human performance can be achieved through education, training, and simulation tools.

The Department of Defense has invested billions of dollars developing substantial expertise in state-of-the-art training and simulation technology. Although this technology has many civilian applications, very little has been transitioned to the civilian sector. Principal barriers to this transition include lack of knowledge about commercial markets on the part of defense contractors, and lack of knowledge about defense technologies on the part of potential customers. The Training and Simulation Technology Consortium, through its varied membership and staff expertise, combines knowledge of commercial markets with knowledge of defense technologies to provide the basis for eliminating many of the barriers to commercialization.

Specific Need: GameShell

A specific example of the need to provide a resource to assist defense based companies in commercializing military simulation and training technology, and in fact an impetus for the creation of the Training and Simulation Technology Consortium, was the attempt to market a software product developed through a Cooperative Research and Development Agreement (CRADA) between the Naval Air Warfare Center Training Systems Division (NAWCTSD) and Dynamics Research Corporation (DRC). NAWCTSD and DRC jointly developed a software tool which enables instructors to enter test questions in a database and quickly generate educational testing games. Transitioning the software to military users was relatively easy. A message describing the software was released and software was provided in response to requests. Commercial distribution of the software was not so simple. DRC was confronted with the issues of locating markets, advertising, pricing, packaging, marketing, distribution, and customer support. The product, GAMESHELL, was the first commercial education or training venture for the company. DRC did not have a commercial marketing group. They did have an employee who had worked in the vocational education market and knew of some software distributors. DRC approached these vocational education product distributors, who agreed to sell the product. These distributors reviewed the product and suggested a price. On this basis, DRC invested in a commercial formulation and packaging of the product. Initially, there were no commercial sales. Soon, DRC recognized the need for expert market research and contracted with a marketing consultant. As a result, DRC restructured the pricing, produced some promotional material, and began sales of the product.

In looking back, it was apparent that the product was launched without the benefit of sufficient market research to properly identify the best customer group and to determine the correct pricing strategy. Because DRC was organized to operate in the defense contracting environment in which a product is developed and delivered to a predefined customer, they were unprepared to address a mass market. The commercial world operates quite differently from DoD. Prices are set by the market place, and a product must be sold to many customers to generate a return on an investment.

Addressing the Needs: Responding to the Technology Reinvestment Program

The expertise and technology which supports military training is viewed by many to represent the state-of-the-art. During the same time period that DRC was marketing GAMESHELL, federal policy officials were examining how best to transfer defense training and simulation technology to civilian applications. It seemed that much of the technology developed for military training could be applied to other areas such as public education and commercial training with tremendous gains in human performance. In fact, the defense simulation and training industry and military agencies were cited in President Clinton's and Vice President Gore's Economic Plan as a national resource to be tapped for refueling the nation's economy (Clinton, 1993). Yet the problem of disseminating this technology through commercialization remained. With defense downsizing, few companies have the resources to acquire the commercialization expertise needed to modify and market the technology in new customer segments. Further, in many cases, companies have a limited number of products and little expertise with commercial applications. Determining the products with commercial potential, identifying markets, deciding how much to invest in commercialization, selecting best techniques for product distribution and marketing -- all require expertise in a wide range of commercialization capabilities.

With the announcement of the Technology Reinvestment Program came the promise for funding assistance in defense conversion and the opportunity to create a new way to deploy federally funded technology. NAWCTSD, with the Army Simulation, Training, and Instrumentation Command and NASA Kennedy Space Center as government partners, teamed with Loral Federal Systems (then IBM Federal Systems), DRC, Analysis & Technology, Dual, Inc, and the University of Central Florida to submit a proposal which would establish a Training and Simulation Technology Consortium to support the commercialization of defense simulation and training technology and consulting expertise. As described in the proposal, the consortium will provide market research expertise to identify potential customers for the DoD technology, matching available DoD technology to the customer requirements thereby applying the technology for commercial uses. Also, the consortium was designed to serve as a resource for those seeking simulation and training technology and expertise by providing information on specific technologies and expertise available in the DoD. The Training and Simulation Technology Consortium was developed to assist both defense suppliers in finding new markets and work with non-defense customers to locate defense simulation and training technology. The consortium will do this by:

- Determining markets which could best use the DoD developed technology;

- Identifying the sources of the technology from among the DoD suppliers;

- Providing a focal point for access to the technology for potential customers in civilian markets;

- Providing DoD industry with product commercialization and consulting services;

- Conducting on-going market research to maintain an up-to-date customer group;

- Becoming a self-supporting organization within three years through membership, marketing or product royalty fees;

ESTABLISHING THE CONSORTIUM

Between the time of proposal submission (23 July 93) and the date that the Consortium learned that their proposal was selected for funding (24 Nov 93), the team members continued to meet to discuss how to implement the proposed concept. With the announcement of the award, efforts became even more intense. The proposal called for the consortium to organize as a not-for-profit corporation with a Management Board comprised of consortium members. The proposal stated that the corporation was to be headed by a salaried executive director, who serves at the pleasure of the management board. (Moving in a New Direction: Training and Simulation Technology Consortium, 1993.) This board was to be comprised of voting members from each organization forming the consortium. The Executive Director would work with a small staff of five professionals and administrative specialists including a Marketing Coordinator, Technology Coordinator, Consulting Coordinator, Administrative Assistant, and Secretary/ Receptionist, plus several consulting experts.

The initial tasks confronting the original team were to:

- actually form the consortium (both the corporation and the relationships of the members);

- hire the staff;

- prepare for and negotiate with ARPA;

- and define ways for organizations to become members of the Consortium.

Organizational Structure

An issue surfaced with the University of Central Florida concerning their role. The proposal stated that the consortium would contract with the Business Development Group in the College of Business Administration at the University of Central Florida (UCF) for human resource management services to include recruitment, selection, training, development, supervision, and benefits coordination. UCF would also serve as the fiscal agent. This was the proposed approach because legal counsel raised the issue of whether ARPA would provide funding to a new entity such as the TSTC or if there were advantages in allowing UCF to administer the funds. Two factors caused the members to change the initial approach. First, ARPA had no objections to contracting with a new entity provided that appropriate procedures were followed, including use of a commercial accounting firm for audit purposes. Second, the members found that by establishing a separate entity, they could more closely approach the structure of a small, entrepreneurial firm and reduce administrative costs.

This decision did create some additional tasks in establishing the TSTC, Inc. The members had to locate a commercial audit firm, find a bank, establish an accounting system, and locate personnel support, including a way for providing employee benefits. These tasks were performed by subcommittees comprised of the initial team members and representatives from their organizations. The budget was also revised by a subcommittee to reflect these changes.

An innovative approach to the personnel support issue was to form an agreement with an employee leasing firm in a co-management role to act as TSTC's personnel and payroll department. Administaff was selected because they offer competitive employee benefit packages, equal employment opportunity assistance, management training, and aid in screening and hiring employees. By adopting this approach, the members were able to minimize overhead costs while providing extremely competitive employee benefits.

Legal Issues

Loral Federal Systems and NAWC-TSD provided legal counsel to help formulate the by-laws of the consortium.

Devising and getting approval for the legal structure of the TSTC was made easier by involving a law firm familiar with the procedures of a consortium. The TSTC was incorporated in Florida as a not-for-profit corporation in April 1994. Legal issues confronted by the members included anti-trust legislation, liability, and conflict of interest concerns. The new Executive Director's experience solved many of these issues. The anti-trust issue of competitors coming together to work collectively became a simple one to address. Legislation supporting the Technology Reinvestment Program and cooperative agreements with ARPA exempts companies from anti-trust legislation. The issue of liability concerns for the Board of Directors was addressed by the purchase of insurance for the Board Members. The third issue of conflict of interest has been somewhat more difficult to address. The issue arises because the TSTC member companies bid for government contracts awarded by the TSTC federal members. Current Department of Defense conflict of interest regulations restrict DoD employees from serving as Board of Director members in an official capacity. The potential conflict occurs when it is perceived that government officials may help companies compete for government contracts by becoming involved in the management of the corporation.

Under the present arrangements, the government members are members of the Consortium and are linked to the TSTC, Inc. through a document called a memorandum of participative cooperation. The government members serve as advisors to the TSTC, Inc.'s Board of Directors by participating on a committee. Unlike Board Members, they will not vote on issues presented to the Board for approval or resolution.

A charter and by-laws were written for the operation of the TSTC, Inc. These documents outline the purpose of the corporation and provide rules for its operations. These documents were submitted, along with Articles of Incorporation, to establish the TSTC, Inc., as a not-for-profit corporation under Florida law. Included in these documents are the in-kind contributions provided by the Charter members of the consortium. These documents were reviewed by all members and signed by all corporate members.

Hiring of Employees

The staff hiring process was handled by a subcommittee of the consortium, with oversight by all members. Advertisements were placed in the Wall Street Journal and The Orlando Sentinel for all positions. Over 600 resumes were received for all positions. The team determined that it was best to hire the director, and then allow that individual to hire the remaining staff. Mr. Michael Walter was hired as the executive director. He began working immediately, even prior to receipt of ARPA funds, to complete the final organizational tasks and to operate the Consortium. In addition to the Director, four other positions were filled.

Working with ARPA

The arrangement being used under the Technology Reinvestment Program to fund the TSTC is a modified Cooperative Agreement. Under a Cooperative Agreement, costs are shared between the "contractor" and the government. Many aspects of the TRP are unlike other forms of government contracting. The Contracting Officer's Technical Representative, or ARPA Executive Agent, for this effort is the NASA Consortium representative, Priscilla Elfrey. The guidance she received for this effort is to pursue new, innovative business practices in the contracting process. She took advantage of the advisory committee and several legal, and accounting consultants that helped guide the TSTC, Inc., to insure that the objectives outlined in the proposal are met. She helped the team organize and assemble the information needed to negotiate a contract with ARPA's agent, NASA. Therefore, little information was missing for the actual negotiations. The TSTC, Inc. was advanced some funding under a modified cooperative agreement to begin operations, while a final agreement was prepared.

Membership Procedures

A committee was formed to determine how companies and other organizations could join the TSTC. The original members all made significant in-kind contributions as part of the requirement for the ARPA award. They also contributed significantly in the proposal development process. Therefore, a strategy for membership was needed that would acknowledge this initial contribution, yet allow for expansion of the Consortium. The committee identified several categories of membership based on the level of contribution. Members joining can receive various levels of assistance, and contribute to TSTC operations in accordance with their membership status.

BEGINNING OPERATIONS

Even before the Executive Director was hired, the Consortium began to work together to identify new applications for Defense Simulation and Training Technology. One of the team members, Bill Jorgensen from DRC, acted as the interim Executive Director of the Consortium. He represented the consortium during meetings with the Los Angeles County Sheriff's Department, the California Commission on Peace

Officers Standards and Training, and AGC Corporation by outlining the capabilities of defense simulation and training technologies. As a result of this visit, the TSTC has organized a panel for I/ITSEC on law enforcement training requirements.

As outlined in the proposal, the consortium will perform tasks in two equally important, general categories: technology commercialization and consulting services. In the area of commercialization, the TSTC will perform two tasks. In the first task, the consortium will endeavor to identify and catalog the major defense training and simulation technologies. Market research will be performed to identify viable marketplaces and to gain insight into the needs of civilian target customer sets. Analysis then will be performed to identify and recommend defense training applications or technologies which can be modified to serve the needs of the new customer. Lastly, research will be performed to identify cost-effective distribution strategies. Work is underway to support this task. Consortium staff are becoming familiar with existing databases of defense simulation and training technology. They have also examined the commercial training market, and the state and local government markets and are developing a customer database, which will help bring new members to TSTC and new non-DoD customers for TSTC members.

The second task was to test the commercialization process on a known product. A specific training and simulation technology or product was identified and the analysis necessary to plan and implement its commercialization was performed. Market research helped identify potential marketplaces, analysis of the technology/product determined appropriate recommended modifications, potential sources of venture capital will be identified to assist in the commercialization effort, and potential product distribution channels will be identified.

In the area of consulting services, the TSTC will perform two tasks. The first will seek to confirm the goals and plans of the consortium for providing training and simulation consulting. The TSTC will identify and catalog the resources available within its membership with specialized training and simulation expertise. Market research will be performed to verify and better understand the target customer markets and their needs. Research will be performed to identify and implement appropriate, cost-effective access and delivery mechanisms for these consulting services. Again, work is currently in progress towards completion of this task. Member companies and government agencies have briefed the TSTC, Inc. staff on their capabilities and supplied them with documentation of their expertise.

The second task in this area targets a worthy, public sector customer (e.g. Florida School Year 2000) and provides consulting services to support that customer's specific training objectives. TSTC, Inc., is currently supporting School Year 2000 by serving as a "red team" reviewer for efforts, and by locating defense simulation and training software which might apply to their training requirements.

Several opportunities to apply certain DoD technologies and training applications are being pursued:

Nurse Job Aid--TSTC is working with the Orlando Regional Medical Center to develop a job aid for new registered nurses to assist them in delivering quality nursing care.

Association Training--TSTC is developing a partnership with Convention Planning Services, Inc. to provide training analysis and design expertise to major trade associations such as the American Dental Association, the American Medical Association and others.

Noise Suppression Technology--TSTC is working with Analysis & Technology and others to commercialize a noise suppression technology for the public telephone component industry.

Secure Wireless Technology--TSTC is working with TSTC members on commercialization of encrypted wireless technology for location based entertainment and resort industry uses.

EVALUATING THE PERFORMANCE OF THE TSTC

Quarterly progress reports will be provided to ARPA which will describe actions taken to achieve success.

The measures of the TSTC's success will be in the areas of:

- number of commercialization projects;

- number of consulting hours provided for customers,

- increase in formerly defense dependent companies' non-DoD business base,

- gains in performance achieved by non-DoD customers through use of these products.

The ultimate measure of performance is the ability of the TSTC, Inc. to achieve self-sufficiency--to operate without federal funds. To become self-sufficient, the TSTC, Inc. must help the defense simulation and training industry commercialize their expertise and products. The TSTC is responsible for finding customers who are willing to invest in the modification or development of products from the DoD provider. This will result in commitment by the customer, financial gain for the provider, and revenue for TSTC to become self sufficient.

A LOOK AT THE FUTURE

American's tax dollars have been invested heavily in training and simulation technologies for our highly skilled military services. This capability in defense systems, technology and know-how are valuable resources, which have great potential to provide training and education to America's students and workers.

The TSTC was established to foster job creation by helping its members commercialize their capabilities and to increase productivity in the civilian sector through use of the defense technology. TSTC connects its members with potential customers to whom solutions can be sold with the involvement and participation of the customer. The eventual product or service can then be productized for a broader market consisting of similar customers looking for the same or similar products.

Fundamentally, TSTC exists to commercialize products or services leading to incorporation of DoD products into the non-DoD markets. The approach of the TSTC is to leverage the tremendous technology investment of the DoD using start-up funding from ARPA, matched by participating corporations, and create commercial markets for DoD products. This approach recognizes the critical need to identify markets, communicate with potential customers, determine their needs, and re-engineer the DoD products to meet those needs. Using an aggressive commercialization and consulting approach for existing DoD products, the consortium will enable corporations to successfully deliver these products to a new market place.

If successful, the TSTC will realize the following benefits:

- DoD investments in training and simulation technology will be leveraged for maximum return by making this technology available and affordable for non-DoD users,

- applications will be tailored to meet customer needs and budgets to ensure that the best training or simulation solution is provided, thereby increasing productivity in the work place,

- opportunities for dual use development will be identified and pursued by industry and government.

The ultimate goal of the Training and Simulation Consortium is to enable delivery of well designed simulation and training products to new markets through well planned and managed market research, matching products to market needs, and productizations (re-engineering to meet market demands) of the products.

The consortium is poised to take the next logical steps to bridge the current gap between DoD developed technology and potential new markets by providing a national focal point for effective training and simulation technology transfer.

References:

Clinton, President William J., and Vice President Albert Gore, "Technology for America's Economic Growth: A New Direction to Build Economic Strength," February 22, 1993

"Moving in a New Direction: Training and Simulation Technology Consortium", Proposal submitted to Advanced Research Projects Agency under the Technology Reinvestment Program, June 23, 1993

MINIMUM ESSENTIAL CDRL REQUIREMENTS FOR SIMULATOR SOFTWARE DOCUMENTATION

Igor V. Golovcsenko
Training System Program Office
Wright-Patterson AFB, OH

ABSTRACT

This paper describes an approach to streamline software data acquisition with recognition of both the contractor's role in technical design development and the Air Force's role in managing requirements. It describes recommendations of an Air Force/ Industry CDRL Corrective Action Team, implementation on recent contracts, and feedback from the simulator community. The goal of the Air Force/Industry partnership was to minimize cost and time for preparation, review and use of documentation while ensuring effective and continued sustaining support through the life cycle of the simulator system.

ABOUT THE AUTHOR

Igor Golovcsenko is Computer Resource System Engineer and until recently Chief, Computer Systems Acquisition Branch, Training System Program Office (SPO), Wright-Patterson Air Force Base. His experience includes 17 years at Aeronautical Systems Center (ASC) and 11 at Naval Training Equipment Center (NTEC), Orlando and Port Washington (Sands Point) where his first assignment was to design and build an analog computer simulation of a T-28 aircraft. He presented several technical papers at Navy/Industry Training Equipment Conferences. At the annual National Aerospace & Electronics Conference (NAECON) in Dayton, Ohio, he has participated as author, session organizer, session moderator, chairman of special sessions, and general papers chairman. He is since 1988 Training SPO representative to the National Security Industrial Association (NSIA) Simulator Computer Working Group, and he was a member of the Air Force/Industry Contract Data Requirements List (CDRL) Corrective Action Team.

Mr. Golovcsenko has an A.B. in Physics from Cornell University (1963), an M.S. in Industrial Engineering & Management Systems from Florida Technological University (1974), and an M.S. in Simulation Systems, University of Central Florida (1987). Born in Budapest, Hungary, he recently returned for a visit after 44 years.

MINIMUM ESSENTIAL CDRL REQUIREMENTS FOR SIMULATOR SOFTWARE DOCUMENTATION

Igor V. Golovcsenko
Training System Program Office
Wright-Patterson AFB, OH

BACKGROUND

Under auspices of Total Quality Management (TQM), a group of Air Force and industry specialists examined existing training system contract data requirements for potential improvements and cost savings. The mission of the Air Force/Industry partnership was to identify and promote implementable approaches to minimize the cost and time required for preparation, review and use of data, but still ensure effective and continued sustaining support through the life cycle of the training system. Application of a Total Quality Process focused on the users and their requirements, analyzed how work was accomplished, and led to the identification of "non-value-added" data items and subsequent recommendations. The team members listed in Table 1 consisted of functional specialists from the Training System Program Office (SPO), Ogden Air Logistics Center (OO-ALC), and training systems contractors.

Table 1. Air Force /Industry Team

LT COL MIKE UECKER - ASC/YW	MR DAVID KUHNS - Flight Safety
MS PEGGY JONES ASC/YW	MR RICHARD RUBRECHT - CAE-Link
MR CRAIG MCCLEAN - ASC/YW	MR ALL EMERSON - ECC International
MR DAVE WELLMEIER - ASC/YW	MR DAN JUCHUM - General Electric
MR HUBERT MERRY - OO-ALC	MR WILLIAM PRITCHARD - Loral
MR KENNETH ACOCKS - OO-ALC	MR PAUL MALIGARY - Hughes
MR RICHARD CARLSON - OO-ALC	MS KAREN BOND - McDonnell Douglas
MR IGOR GOLOVCSENKO - ASC/YW (temp)	

The group, known as the Contract Data Requirements List (CDRL) Corrective Action Team, was chartered by the Air Force/Industry TQ Steering Committee to take a "clean sheet" approach in making its recommendations. This approach allowed the team to step outside the paradigms that encourage business as usual and to attack the problem with relative lack of constraints.

The team examined the CDRL requirements on both the Special Operations Forces (SOF) Aircrew Training System (ATS) and the C-17 ATS, which represented the most recent large programs being managed by the Training SPO. The B-52 Contractor Logistics Support (CLS) CDRLs were also studied to capture the full spectrum of CLS efforts. The team examined commercial training system buys, and the data purchased by the airlines from training system contractors. In addition, experts from the various functional disciplines were either consulted or temporarily assigned to the team to share their knowledge and ensure a quality product.

The team determined that major reductions in paper submittals could be made if a new way of business can be initiated. This new way of doing business can be characterized in three ways:

a. Trust: Both the Air Force and industry realize that training system contracts are "win-win" or "lose-lose" propositions. Therefore, the Air Force trusts that the contractor is honest and trying to develop and maintain the product asked for. At the same time, the contractor trusts that the Air Force is really part of the development team with a real need for timely, accurate and complete information.

b. Delivery of data only when and where it is needed: The team recommended that the formal delivery of interim data be replaced by on-line access or electronic delivery of much data. This approach eliminates the contractor formal management review process and speeds up data delivery to the responsible Air Force team member.

c. On-line access: This would enable Air Force/Industry interaction at a much lower level and in a more timely way. Networking data would allow the Air Force to read, review and comment (but not change) without unnecessary paperwork, and also reduce miscommunication and delays in the overall program.

FUNCTIONAL AREAS

The corrective action team used a Delphi approach to determine a set of recommended solutions. The following functional areas or disciplines were broken out, and functional specialists were tasked with analyzing and recommending ways to eliminate "least value-added" data requirements:

a. Hardware. Engineering drawings were the most significant hardware data item in terms of cost and amount of delivered pages.
b. Software. Thirteen data items were examined.
c. Courseware. This area addressed Instructional System Development contracts.
d. Management. Data dealing with cost, schedule, meeting agendas and minutes.
e. Acquisition Logistics Support. This area addresses support data requirements for acquisition logistics management and long-term supportability.
f. Sustaining Support. Data items that support follow-on and modification contracts.
g. Test. Test documentation.

h. Facilities. Data on facility requirements for the system to be fielded.

TYPES OF DATA

The initial meetings provided opportunities for the sharing of information on how data is purchased and used by the various organizations within the Training SPO and Ogden ALC. The industry members of the team described the data they provide to their commercial customers with commercial training systems. It became apparent that the Air Force buys three types of data:

a. **Perishable.** Data in short-term use required during development to manage the effort. This type includes management reports, conference agendas, and schedule reports, with primary use being to determine program progress. Perishable data is obsolete soon after it has been gathered.

b. **Evolving.** Data which matures during the development phase and defines the characteristics of the training system and system components. This type too is used by the Air Force to ensure that a contractor is progressing toward successful accomplishment of the program's technical objectives. However, in its final form, this data is also in use for maintaining and updating the training system. Software design documentation is an example of evolving data.

c. **Permanent.** Data required for operation, maintenance and support (OM&S) of the training system after it is fielded. This data, (for example, technical manuals and drawings) is generated toward the end of the design phase, and then maintained current through the life of the program. Permanent data is in use to support major modifications and recompetition, as well as day-to-day operations.

PURCHASE OF DATA

When the team analyzed a representative sample of the data purchased by the Air Force in conjunction with aircrew training systems with Contractor Logistics Support (CLS), several observations were possible:

a. Data delivered during the development phase is normally obsolete (at least a month old) when received by the Air Force. This delay is caused by a number of factors, including required contractor internal reviews, reformatting, the data collection process, etc. This obsolete data is then reviewed by the Air Force (typically a 30 to 45 day process), and returned with comments. Quite often these comments are invalid because the contractor has, during the two-month span, already corrected the problem or changed the design. This miscommunications results in frustration for both the Air Force and the contractor.

b. The Air Force typically buys an entire set of documentation when a data sample would suffice to check for accuracy and methodology.

c. The Air Force buys quantities of data that need only to be looked at, and has no further use (e.g., status reports).

OVERALL RECOMMENDATIONS

The corrective action team reviewed a total of 102 data items. Based on the research accomplished, and using the new approach, the team recommended that only 58 of these data items be retained, with an even greater reduction in the number of deliveries due to the elimination of interim submissions. Details are contained in Reference 1.

PRINCIPLES OF STREAMLINING

The corrective action team completed its analysis of minimum essential requirements for software. In this area, the team proposed significant changes in response to the basic goals for minimal data, namely execution of Air Force oversight responsibility during development and effective life cycle product support

The proposed approach to streamline and standardize software data acquisition was based on the following principles:

a. Emphasis of **contractor's** role in technical design development, as distinct from the **Air Force's** role in managing program design/development requirements.

b. Reliance on contractor management systems, automated data capability, and real-time data access to support Air Force data requirements.

c. Elimination of excessive specification review, approval and authentication process.

d. Deletion from the CDRL of contractor generated documents that do not require formal Air Force review and approval.

e. One-time only formal delivery of support documentation.

SOFTWARE DATA ITEMS

The corrective action team examined 13 software data items in DOD-STD-2167A, Defense System Software Development. They are listed in Table 2.

Table 2. Software Data Items

DID NUMBER	DID TITLE
DI-MCCR-80030A	Software Development Plan (SDP)
DI-MCCR-80025A	Software Requirements Specification (SRS)
DI-MCCR-80026A	Interface Requirements Specification (IRS)
DI-MCCR-80027A	Interface Design Document (DD)
DI-MCCR-80012A	Software Design Document (SDD)
DI-MCCR-80029A	Software Product Specification (SPP)
DI-MCCR-80013A	Version Description Document (VDD)
DI-MCCR-80014A	Software Test Plan (STP)
DI-MCCR-80015A	Software Test Description (STD)
DI-MCCR-80017A	Software Test Report (STR)
DI-MCCR-80018A	Computer System Operator's Manual (CSOM)
DI-MCCR-80019A	Software User's Manual (SUM)
DI-MCCR-80021A	Software Programmer's Manual (SPM)
DI-MCCR-80022A	Firmware Support Manual (FSM)

RECOMMENDATIONS

Of the 13 software data items, 5 were recommended to be retained. Another 4 were recommended to be delivered as CLS manuals under a single data item, with some tailoring. Table 3 summarizes these recommendations. It is followed by detailed explanations of the rationale used.

Table 3. Summary of Recommendations

RETAIN	TAILOR	DELETE
Software Devel Plan	Comp Sys Oper Manual	Software Req Spec
Interface Design Doc	Software User's Manual	Interface Req Spec
Software Design Doc	Software Progr Manual	Softw Test Plan
Software Prod Spec	Firmware Suppt Manual	Softw Test Descr
Version Descr Doc		Softw Test Report

Software Development Plan (SDP)

It is generally recognized that planning and scheduling of resources is essential for project success. The SDP describes a contractor's plans for conducting software development. Used by the contractor team throughout development as the agreed-to approach, the SDP also provides the Air Force with insight into the organization(s) responsible to do the job and the methods/procedures to be followed by this organization. The SDP will be the primary contractor planning document governing the development of software. A preliminary SDP may be submitted as part of the selected contractor's proposal, and it becomes the working SDP upon contract award. The program office ensures that the SDP is updated as necessary to remain current throughout the software development cycle and ensures that development activities are managed in accordance with the plan.

Software Requirements Specification (SRS)

This document establishes an allocated baseline for software and defines requirements for software to be designed. The team recommended deleting this data item from the CDRL. The contractor will then (as before) be tasked to conduct activities required by tailored DOD-STD-2167A in accordance with the Statement of Work. These tasks will (as before) include activities necessary for software requirements analysis. The development contractor will be required to allocate system requirements to software, define engineering requirements for each Computer Software Configuration Item

(CSCI), establish traceability to the System Specification, and maintain control of the allocated requirements through the contractor's own internal configuration control. Software requirements will be controlled as a **"developmental baseline"**. However, delivering the SRS as a formal document was not considered necessary to accomplish the activities of software requirements analysis. The rationale is as follows:

a. There is one SRS per CSCI. As the Training SPO requires no formal CSCI acceptance test, the need for an Air Force controlled allocated baseline is not evident. Only the **functional baseline** and the **product baseline** require formal government control. Figure 1 illustrates this view of configuration control during development.

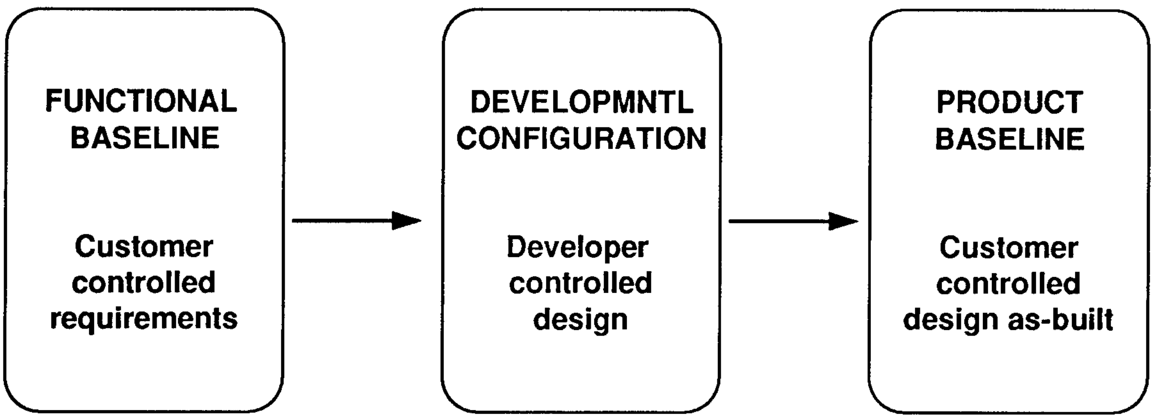


Figure 1. Configuration Control

By requiring a separate software requirements document, we theoretically document functional requirements twice, once in the Item Development Specification and again in the SRS. Probably the more common result is to defer the details from the Item Development Specification to the SRS and then to fail to capture the heart of the required action, which was to produce a system level "function". The software function itself becomes an abstraction; for example, "react to an interrupt or a bit in an I/O port".

b. Air Force baselining and maintenance of an SRS involves formal reviews, authentication, configuration control boards, and Engineering Change Proposal (ECP) processing all costly to implement. Changes impacting these specifications once under government control must be through the laborious ECP process. Heavy change activity all through development leads to frequent Class I engineering changes which in turn leads to probable schedule impact. The data item, DI-MCCR-80025A, stipulates each required capability to be identified in a uniquely numbered paragraph, and that the SRS include details of internal interfaces for each individual capability, down to the associated internal data elements. Providing these internal data interfaces in a requirements specification can substantially affect the cost and schedule of the design effort. The required capabilities, their internal interfaces, and each data element must be translated directly into the top-level design. As the developer's design group starts its top-level design, it may decide on a different partitioning of the software. Because this affects the data elements identified in the SRS, a formal agreement with the Air Force is required to make that change in the requirements specification.

c. Mapping the software architecture of a system into a structure imposed by the SRS is difficult. Some principles and constructs of modern programming languages such as Ada do not well map to the definitions in DOD-STD-2167A and DI-MCCR-80025A. Principles and constructs of design methodologies such as object-oriented design and structural modeling also do not map too well to the definitions in DOD-STD-2167A and the data requirements in DI-MCCR-80025A. In previous "standard" procurements, architectural definitions were not considered and developed until after software requirements were specified. However, for several recent programs, software architectural requirements were defined in parallel with the software functional requirements.

d. The SRS is typically not maintained after simulator delivery.

For these reasons, it is preferred that the SRS not appear in the CDRL, and that contractors tailor the SRS format to internal use for allocating and controlling the internal development configuration. This internal specification would be made available to the customer upon request via the Data Accession List.

The National Security Industrial Association (NSIA) Working Group for Simulator Computers recommended that if formal CSCI testing is to be accomplished separately from system integration and testing, an SRS should be required. For that type of procurement, the NSIA working group recommended the following paragraphs be tailored out of DI-MCCR-80025A:

10.1.5.3 CSCI Internal Interface

10.1.5.4 CSCI Data Element Requirements

In addition, the sentence referencing inputs and outputs of capabilities in paragraph 10.1.5.2.1 should also be deleted.

Interface Requirements Specification (IRS)

The Interface Requirements Specification (IRS) is used to specify requirements for interfaces between one or more CSCIs and other configuration items. Use of an IRS is typically in combination with one or more SRSs. This data item should be deleted from the CDRL when the SRS is not required. When deleted, the contractor would tailor the IRS for internal use, as the SRS.

Software Test Plan (STP), Software Test Description (STD), Software Test Report (STR)

The Software Test Plan, Software Test Description, and Software Test Report are dependent on the treatment of the SRS. Therefore, these data items should not be required when the SRS is not required. This level of testing should be accomplished as part of the system-level training system/simulator acceptance tests.

Software Design Document (SDD)

The Software Design Document (SDD) provides engineering data and technical description essential to understanding the design. Developed during preliminary and detailed design activities, the final version of this document becomes integral to the Software Product Specification. Formal delivery of this data item could be in conjunction with the Software Product Specification only, whereby no "formal" Preliminary Design Review (PDR) and Critical Design Review (CDR) deliveries would be required.

Approval is the act of acknowledging legal responsibility by the government for the accuracy, adequacy and completeness of the data. Therefore, formal PDR and CDR deliveries require formal Air Force review and approval, requiring the Air Force to assume legal responsibility for adequacy, accuracy and completeness of interim SDDs. The procuring activity has general access to the contractor's working data generated to document compliance with task requirements. These data can be reviewed at the contractor's facility.

During system development, the SDD evolves under the contractor's developmental configuration control. Incremental versions of SDDs are working drafts of future deliverables. A key question here is not whether this document is required, but whether it is necessary to have this **interim** documentation **officially delivered** to the Air Force. The corrective action team's recommended approach was to avoid formal delivery of interim paper products and to select alternate methods to access this data and communicate the information. The following methods enhance the Air Force's oversight process:

- Direct and frequent contact with contractor personnel
- Data Accession List items
- Access to the contractor's working data
- On-line electronic information system
- In-plant access to contractor's work station
- Interim SDDs on floppy disk

Using data informally obtained as it is generated, along with updated information received through technical interchange meetings, is usually more timely than reviewing a formal SDD submittal. The additional time required to formalize SDD submittals through an editorial and production process, and then to formally review them, would tend to make this product "out of sync" with the on-going development effort. Design information tends to be volatile during software development, and the document review/update cycle tends to usurp the contractor's effort from the primary activity.

Traditional PDRs and CDRs have been preceded by volumes of CDRL-mandated documents and characterized by large audiences receiving functional/subsystem briefings over a 3-5 day period. In contrast, incremental PDRs and CDRs that focus on actual contractor performance based on mutually defined demonstration milestones and exit criteria promote timely and informed team review and decision-making at the subsystem level. System level reviews then summarize cumulative efforts of incremental, small, multifunctional team reviews conducted throughout the design phase. "Summary" PDRs and CDRs emphasize integration of subsystems and satisfaction of overall user/operational requirements,

Software Product Specification (SPS)

Tailoring the SDD involves evaluating each requirement in DI-MCCR-80012A to determine its suitability and cost effectiveness for a given program. For DIDs, requirements may be deleted or partially deleted (but not modified). Tailoring is intended to reduce the number of requirements levied on the contractor. Tailoring of DID contents is specified in the CDRL (DD Form 1423), not in the Statement of Work. One method used by the C-17 Maintenance Training Device (MTD) program is as follows:

Transfer details of low level design to source code listings. This tailoring option is to document portions of the detailed design in the source code itself, not the SDD. The effect is to reduce the size of the SDD, move the documentation closer to "self documenting" code, reduce the software Physical Configuration Audit (PCA) workload, and make the software more supportable. Changes made in the program often do not appear in the documentation. Putting elements of the documentation in the source code can lead to better maintenance and ensure availability of documentation to the software maintenance engineer. This is accomplished by transfer of detailed design information into module header comment blocks, thereby augmenting the typical "prologue" comment statement with information then not duplicated in the SDD. Forwarding some of the SDD information into the source code may also make sense from the viewpoint that the SDD would not be formally delivered until training system and product specifications were also delivered -- the SPS contains the SDD *plus* the source code. Forwarding requirements to the source code would maximize the utility of the information provided, reduce the need for cross-referencing between design descriptions and code, eliminate redundancy between support documents and the code, and reduce the software Physical Configuration Audit (PCA) workload by at least 25%.

Version Description Document (VDD)

Each submittal of this document identifies and describes a new revised version of a CSCI.

Computer System Operator's Manual (CSOM), Software User's Manual (SUM), Software Programmer's Manual (SPM), Firmware Support Manual (FSM)

The information specified to be delivered in these documents is extremely cognizant to the operational support of a trainer. However, this information is more appropriately located in the Technical Orders/Manuals, which are more often used for the support. Collocating this information in one document allows more quick and easy reference. The recommended data item, TM-86-01/T, "Technical Manuals/Commercial Literature", specifies the development and acquisition of Contractor Logistics Support (CLS) manuals and contractor data and it was developed by the Training SPO Logistics Division (ASC/YWL).

TM-86-01/T defines the Contractor Logistics Support Manual (CLSM) Contract Requirements. The term "CLSM" used throughout the document denotes commercial manuals and other forms of contractor data. CLSMs delivered by the contractor will be formatted in accordance with best commercial practices. Section 3 contains specific requirements for manuals, including requirements for a Computer System Operator's Manual and a Software User's Manual using DI-MCCR-80018A and DI-MCCR-80019A as guides. The CSOM may reference COTS manuals to be provided. TM-86-01/T has other requirements as well for maintenance information regarding software and firmware.

IMPLEMENTATION

The Simulator for Electronic Combat Training (SECT) and F-15/F-16 Unit Training Device (UTD) are two recent programs implementing the approach. Table 4 summarizes the software data items for these programs.

Table 4. CDRL Implementation

DID NO.	DID TITLE	SECT	UTD
		(CDRL Sequence No.)	
DI-MCCR-80030A	SDP	A0030	C020
DI-MCCR-80025A	SRS		
DI-MCCR-80026A	IRS		
DI-MCCR-80027A	IDD	A0028	A018
DI-MCCR-80012A	SDD	A0026	A015
DI-MCCR-80029	SPS	A0029	A020
DI-MCCR-80013A	VDD	A0027	A017
DI-MCCR-80014A	STP		
DI-MCCR-80015A	STD		
DI-MCCR-80017A	STR		
DI-MCCR-80018A	CSOM	A0044	B001
DI-MCCR-80019A	SUM	A0044	B001
DI-MCCR-80021A	SPM	A0044	B001
DI-MCCR-80022A	FSM	A0044	B001

Both programs have now completed Critical Design Review (CDR). For SECT, availability of SRS and IRS are via a Data Accession List (DAL), while the UTD contractor has no internal SRS/IRS, and requirements are tracked directly from the Prime Item Development Specification (PIDS). The UTD CDRL tasks the contractor to produce a requirements traceability matrix for the software requirements "from the Government approved system specification to the PIDS, the software design documents/interface design documents (SDDs/IDDs) and the code documented in the software product specification(s)".

For the UTD program, the SDD available at each major design review (PDR, CDR, EDR) is a working document not yet "formalized" at the time of the review. The SPO computer engineer maintains frequent contact with the contractor's software manager, and the SPO obtains draft versions of design documents by e-mail. Direct SPO access to a contractor workstation is soon to be implemented.

FEEDBACK FROM SIMULATOR COMMUNITY

The two programs which implemented minimum essential CDRL requirements are now in the coding phase. Complete feedback will not be available until acceptance tests. Although at this point in time not all questions can be answered, interim feedback suggests several concerns regarding the following aspects of the recommendations.

<u>Deletion of Requirements Specifications</u>.

Deleting these specifications raises concern about enforcing the development process. Although nothing in the recommendation relieves contractors from developing the information in this document, there is also nothing to persuade them to comply with the intent of the contract when no product is to be delivered. When pressed, contractors may be tempted to cutting corners and their development process would deteriorate.

Maintaining requirements traceability is vital. It not only provides evidence that the design and implementation satisfy requirements, but it also serves as a mechanism to measure requirements volatility. Requirements volatility is identified as a leading cause for latent defects in systems. Whether the problem is caused by a lack of understanding by the contractor, or other reasons, this situation must be adequately managed.

The elimination of formal delivery of the SRS/IRS does not eliminate the need for requirements traceability. The use of tools to maintain the requirements traceability data should be required. The type of tool and the utility of the data should be described as part of the software development environment.

The Software Development Plan should define the requirements process. It should be a life long document signed up to at source selection. It should flow down to subcontractors and contain extensive explanation of requirements development and processes. Poor contractor performance in this area should be penalized.

Deferred Delivery of Software Design Document.

There are reservations that design information will not be presented for approval at design reviews and the perception that the SDD is purely a maintenance document. During system development, the main purpose of the SDD is as a guide to coding the software. It communicates something useful between engineers, serves as a memory aid for the software engineers, and is a repository of the software philosophy, or "architecture". Design representations should be used at the design review and maintained in the Software Development Folders. The design representation may be textual or graphical, but must be adequate to assess the completeness and consistency of the design for the specific subsystem(s) being reviewed.

The SDD is valuable to the customer as a process control document. It indicates whether the system is being (or will be) coded on the fly, or is following a rational plan.

However, has delivery of interim SDDs the intent to measure contractor progress? Delivery of "formal" SDDs 30 days prior to a design review does not appear to be the best measure, as SDD "production" requires several months, and data in "formal" SDDs lags by several months the actual contractor progress.

SUMMARY

A key aspect of this proposal is elimination of the allocated baseline for software requirements and the substitution of a developer controlled baseline. Using a requirements traceability matrix, requirements may be traceable from the government approved system specification/item development specification, through a contractor internal software requirements specification, to the software design documents and the code that's documented in the software product specifications. Performance-oriented specifications at the system and subsystem levels allows the contractor to perform the highly iterative process of design, without being stopped or delayed by the approval of government for each iteration via the laborious and costly ECP process.

The preferred approach
- -requires only essential data for government tracking during development
- -minimizes government involvement in managing the development process
- -emphasizes contractor accountability for development programs and their internal controls

As government/industry relationships improve through effective teamwork, government and industry team members may concentrate on their respective roles that allow industry a design flexibility to meet the government derived requirements.

Acknowledgments. Thanks to the following individuals and groups for their valuable ideas and perceptive comments:

Mr. Dave Wellmeier, Mr. Mark Adducchio, Capt. Norm Tucker, Capt. Don Duckett, Ms. Suzanne Solarek, and Maj. Frederick Swartz of ASC/YW.

Mr. Bob Epps, Mr. Dick Rubrecht, Mr. Paul McMahon, and Ms. Lori Pajerek of CAE-Link.

Mr. Chips Lanier of SAIC, Mr. Jim O'Day of Hughes, and Mr. John Hannah of Information Spectrum.

The NSIA Simulator Computer Working Group.

REFERENCE

1. Air Force - Industry Contract Data Requirements List Corrective Action Team, Final Report, 4 October 1991

EFFECTIVE SELECTION AND USE OF CONFLICT SIMULATIONS (WARGAMES) FOR OPERATIONAL TRAINING OR CAMPAIGN ANALYSIS

Squadron Leader Patrick Beautement MSc, PGCE, RAF
Operational Doctrine and Training Element of the
Air Warfare Centre, Royal Air Force, Cranwell, United Kingdom.
Tel: UK (400) 261201 Ext 6443. Fax: Ext 6539.

ABSTRACT

1. This paper will cover the reasons why conflict simulations (better known as wargames) are used, the types of wargames that exist, how wargames are selected, how they need to be set up and what is required to make the best use of them. Some of the more common myths in wargaming will be dispelled. The impact of future technologies will also be highlighted.

2. Wargames (including models/simulations) have, in general terms, been used mainly for both analysis and training. This paper looks at the context in which gaming is carried out and asks "What types of wargames are available?". The games and simulations available fall into many categories depending on the level of interaction (political to tactical) the style of 'play' and the type of execution (eg manual or automated in some way). This paper considers what wargame choices are available and what selection criteria should be used. Above all, there should be a clear need for a wargame and the game selected should fulfil that need. Also considered in the paper is the impact of new technologies such as synthetic environments and inter-model protocols like ALSP.

3. Setting up a game correctly involves considerations beyond simply the game itself: ie, the selection of equipment and staff, data, rules and scenarios. Once a game is provided, using the game effectively involves further effort. How is the game to be used? Whether the game is to be used for training or analysis, seminar directors, ie subject experts, will be needed to interface with the users. Interpretation is a tricky business and needs to be done with care, as is deciding on the criteria by which "success" or "failure" is to be judged.

4. Overall, the paper will inform readers about wargaming issues and provide methodologies for effective selection and use of wargames.

Key Words:

wargame selection conflict simulation operational training campaign analysis

---

NOTES ON THE AUTHOR: Squadron Leader Patrick Beautement is currently the Wargame Systems Specialist at the Operational Doctrine and Training Element of the AIR WARFARE CENTRE, RAF Cranwell, responsible for running the wargame "TAC Thunder" (used as an "exercise driver" on Battle Staff Training Courses), and for advising RAF staff on wargaming matters. He has been working with computers and computer simulations since he took his Masters degree in Intelligent Systems and Neural Networks before joining the RAF. More recently he has been involved in producing training systems for the Tucano, Tornado and Buccaneer aircraft, as well as managing training design for ground electronic equipment.

EFFECTIVE SELECTION AND USE OF CONFLICT SIMULATIONS (WARGAMES) FOR OPERATIONAL TRAINING OR CAMPAIGN ANALYSIS

Squadron Leader Patrick Beautement MSc, PGCE, RAF
Operational Doctrine and Training Element of the
Air Warfare Centre, Royal Air Force, Cranwell, United Kingdom.

INTRODUCTION

1. This paper will consider why conflict simulations are used, how to select them and how to get the best out of them. But, before I go on to consider these issues it is worth noting that I will be using the words 'conflict simulation' and 'wargame' (to include modelling and simulation) interchangeably. Wargames get 'bad press' in some circles as they do not sound like serious work. However, I would assure the reader that wargames are much more about war than game. As my perspective is military wargaming, I tend to use terminology relevant to that area, however, the selection criteria and guidance on effective use given below are applicable to all areas of work.

2. This article, then, will cover the reasons why wargames are used and, before listing what sorts of games are available, will dispel some of the more common myths about wargaming. The criteria to be considered when selecting games will then be detailed followed by some examples of wargames in current use. Next, the paper explains how games should be set-up and what is required to make the best use of them. Before discussing the future of wargaming, the paper considers some of the technology issues currently in the news. Finally then, I shall suggest some future uses of wargaming.

WHY WARGAME?

3. In essence, gaming is about "investigating the processes of combat", be that combat on battlefields or in other areas of human conflict. Consequently, gaming can be used as an "organising and exploratory device" leading to insights and understandings which would not be apparent unless the gaming occurred. The level of study, be it strategic, operational (ie theatre-level) or tactical, must be considered, as must whether pre-, during or post event issues are to be of concern. Also, are training or analysis tasks to be done? Table 1 shows how these main factors determine the type of gaming activity to be carried out.

		Pre-event	During Event	Post-event
Training	Strategic	Campaign planning	Possible settlements	Conflict studies
	Theatre	Battle-staff training	Campaign analysis	Conflict studies
	Tactical	Mission planning	Mission rehearsal	Mission debriefs
Analysis	Strategic	Doctrine development, Procurement issues	-	-
	Theatre	Force structures	Vulnerabilities	-
	Tactical	Trials and tactics	Loss assessment	Ops analysis

Table 1

4. Wargaming tends to be used mostly for either training personnel or for analysing situations. The reasons for this clear split are as follows:

a. ANALYSIS. In analysis, the purpose is to investigate, validate or prove a concept or plan. In this mode, the parametric scenarios are used to predict outcomes, and as such, are concerned with the product of the game; be it success, failure or some absolute outcome. For the product to be credible, there has to be a strong emphasis on the formulae used by the game to produce the results and on the input data. The wargame must mirror reality and there can be no suggestion of selecting data to produce a set result!

b. TRAINING. In training the purpose is to create a dynamic situation in which to instruct potential "Battle Staffs" or evaluate their performance. Because of this, the emphasis is on the processes that the staffs go through to achieve their goals, and their performance is evaluated against more subjective criteria than in analysis. As far as the students are concerned, as long as the game produces "acceptable" results, the method by which this is achieved is irrelevant. In fact, wargame managers may change data to produce set problems and outcomes for the students to deal with.

Both these types of wargaming **can** be carried out pre- or post event and in all scenarios from tactical to strategic. However, analysis tends to be carried out post event and training pre-event.

MYTHS ABOUT WARGAMING

5. Before I discuss wargaming in more detail, I wish to dispel some of the more common myths about wargaming:

a. MYTH 1 - WARGAMES ARE TOYS. As I mentioned at the start, the words "war game" suggest a trivial, even childish, pursuit. However, in 1824, when the Chief of the German General Staff, General von Muffling, was shown von Reisswitz's "Kriegsspiel" he said, "This is not a game, this is [training for] war!". Von Muffling quickly realised that wargaming could provide a vehicle for examining the [modern] principles of mobility and firepower as well as allowing an exploration of the problems created by the military and political situations of the time. In short, wargaming would assist staff in understanding the demands which would be imposed by a future war. Certainly not a trivial achievement, and still true today!

b. MYTH 2 - WARGAMES = COMPUTERS. When talking about wargaming it is usually assumed that computer-assisted wargames are being discussed. However, it is possible to have very successful manual wargaming. The "Global Wargame" (a strategic politico/military game played at 4* level annually at Newport, Rhode Island, USA) is a good example, where the use of computers would be both limiting and inappropriate. For "training" of this type, it is the conceptual fidelity in the players mind which is of paramount importance and computers are not introduced unless they assist in this process.

c. MYTH 3 - ANYONE CAN RUN A WARGAME. Anyone could, in theory, run a wargame, however, not everyone can run a wargame effectively. Successful interpretation of the results takes skill and experience, close knowledge of the "quirks" of the game (they all have them), a deep subject knowledge and appropriate experience. The old maxim "garbage-in-garbage-out" applies as much to the wargamers as it does to the game itself.

d. MYTH 4 - WARGAMES ARE PREDICTIVE. This is a difficult area. It is clearly possible to use wargames to aid in decision making, evaluation of tactics and strategies and assessment of equipment. However, it is not possible to use games to predict exact outcomes; at best relative values or probabilities can be evaluated. This was clear from the assessment of Gulf War casualties: where from 0.03% to > 25% was predicted. Even statistical analysis needs to be treated carefully. Assumptions, simplifications, limitations and exclusions must all be carefully noted.

e. MYTH 5 - WE'RE IN CONTROL. Well, we (the users of a game, especially a computer-assisted one) ARE NOT (really). We are at the "mercy" of the rules and probabilities "programmed" into the game. A big change in the results can be due to a small change in these rules introduced, say, in a software update. It is also possible to fall foul of data acquired from external sources which may contain generalisations. CAVEAT EMPTOR.

f. MYTH 6 - GOOD GRAPHICS MEANS A BETTER GAME. You can be sure that good graphics attracts purchasers. However, good graphics usually means more expensive hardware, not necessarily a better game. Good "user interfaces" are usually graphical, but if your users don't touch the computer, paying for this may be wasted money.

TYPES OF WARGAMES

6. So what types of wargames are available? The games available fall into many categories as follows:

a. BASED ON SCENARIOS. The games based on scenarios fall into three types:

(1) GENERIC. Generic games are set in mythical scenarios. Their value is that they home in on the procedures and strategies being used, without the distraction of arguments about the "reality" of, say, imaginary enemy actions. However, players can waste time becoming familiar with the generic scenario as they cannot use much of their general knowledge. In short, generic games are best for high level or novice training where a great deal of detail is not required.

(2) SPECIFIC. Conversely, specific wargames are based on reality. They study the effects of known doctrines in real politico/military contexts. Integrity of results can be taken for granted as real values for equipment performance are being used. Players can not only use their general knowledge to good effect, but can also have their knowledge improved through the wargaming process. However, arguments can develop about the validity of the scenario and about the actions taken by different countries. So, specific wargames are best suited to theatre level training downwards.

(3) HYPOTHETICAL. Hypothetical wargames deal with extrapolations of known situations and allow the exploitation of "what-ifs". New or experimental equipment, tactics or overall strategies can be evaluated. However, the danger of this type of game is that too much credence can be given to the outcomes. At best, trends and relative results (not absolutes) can be compared.

b. BASED ON TYPE OF PLAY. There are variations of games related to the type of play used as follows:

(1) FREE OR RIGID PLAY WITH SEMINAR OR SYSTEM RULES. In free play an umpire arbitrates the results of conflict based on their extensive military experience, whereas a rigid play system uses a (usually

complex) set of rules to determine results. The play can be seminar (discursive) or "system" with structured moves.

(2) ONE, TWO OR MULTI-SIDED. One-sided games (against a fixed enemy or red team) allow control of the play to be retained. However, the "enemy" in a computer-based game is the programmer and it becomes too easy to "play the game" and not the war. In two- or multi-sided games the outcomes are more unpredictable, particularly if the teams involved are not closely matched.

(3) OPEN OR CLOSED. In an open game, all players can see all of the action. This format tends to be used for planning games where it is important to discuss enemy options as the play proceeds. Closed games (where one side sees no more of their opposition's moves than they would in reality) tend to be used for training.

(4) CYCLIC OR "REAL TIME". Most training games tend to be played cyclically where a "frozen midnight" is used. At the "game stops" players have time to assess the situation and make their decisions knowing that the enemy play is suspended too. This is unrealistic. Much better is to use a "live day" (where "moves" are made concurrently) to keep the players under pressure while they plan. Even in analysis, using the "live day" can give an insight into the time constraints of a situation.

(5) DETERMINISTIC OR PROBABILISTIC. Deterministic games tend to be "scripted", producing an unrealistic, unreactive form of play. Nevertheless, control of the outcomes is retained, usually at the expense of flexibility. Innovative thinking is not rewarded (or punished!). In a deterministic game, much more effort has to be expended in setting up, maintaining and retaining consistent rules as all eventualities must be programmed in. A probabilistic game produces slick, realistic and reactive play which can deal with unforseen actions by either side, where knock-on effects and interactions may have significant sway. A probabilistic game can automate many processes, for example: the selection of targets, based on a realistic assessment of the actual current threat.

7. All the types of games mentioned above can be executed in 2 main ways; either manually or automated in some way:

a. MANUAL. At the simplest, manual wargames have a fixed "pink" (an answer sheet), determined in advance, which dictates the outcomes. The lack of reactive feedback to inputs makes this kind of game of limited value. More sophistication can be added by dice throwing and the use of "tables of rules" based on previous outcomes. The main problem with a manual system is that determining outcomes for many engagements becomes time-consuming, especially for a large game. However, in a manual game control of results is maintained because the arbitration process is explicit and open.

b. AUTOMATED. Automation usually involves computer-assistance which improves wargaming by allowing faster production of results and increased consistency. Note however that, as yet, there is no such thing as a full computer wargame, only computer-assisted ones. The illusion of complexity is easily achieved in computer-assisted systems and then the danger arises of giving too much credence to the results.

SELECTING WARGAMES

8. So, assuming that wargames are considered to be of value, how are the right ones to be selected? Firstly, it is clear that more than one game will probably be required. Several games may be needed to provide the range of capabilities necessary to cover the different levels shown in Table 1 above. The games selected should be complementary, the strengths of one complementing the weaknesses of the others, with as much compatibility between the games as possible. Above all, there should be a clear need for wargames and the game selected should fulfil that need. However, that "need" (which will usually be to investigate "corporate" decision making processes) must be defined before wargame selection. Games should not be selected, say, just because it is available at little or no cost from another user. So, a number of points need to be considered as follows:

a. WHAT CHOICES ARE AVAILABLE. In choosing new games to meet future needs the question, "What choices are available" must be asked. Is there a definitive list of wargames, their suitability and effectiveness available? Unfortunately, the answer is not really. Many new developments in wargaming are underway, involving technologies such as so called "expert systems" and "artificial intelligence". Consequently capabilities are changing all the time. The Catalog of Wargaming and Military Simulation (1) is one source of information, but the best approach is by talking to existing wargaming users and by seeing games in action.

b. WHAT SELECTION CRITERIA SHOULD BE USED? There is also no established list of assessment criteria for wargames. An MPhil (2), published in the summer of 92, by Sqn Ldr Alan Burton, working at Edinburgh University under the internationally recognised expert Prof John Erickson, gives an up to date overview of some of the selection issues that should be considered. These points are expanded below:

(1) BASIC SELECTION Method. First, consider the type (training or analysis), level of task, scenario type and timescale of the game to be carried out using Table 1. Now consider all the selection issues mentioned in para 6, annotating those relevant to your selected task. Annotate the dependencies in a table. Consider carefully what would be "appropriate" uses. Now evaluate all the factors mentioned below. In this way you will home in on the games suitable for further investigation.

(2) THE OPERATION OF THE GAME. As discussed in paras 6 and 7 there are many types of game and obviously these issues should be considered when selecting a game. However, one of the most crucial issues to be considered when selecting games is how the game operates. This factor not only affects the data and equipment required, but also the quality, experience and number of staff needed. For example, most wargames contain no automatic planning (TAC Thunder is a notable exception to this). Consequently, large teams of support personnel are required to issue orders to all the sea, ground and air units involved. If your aim is to involve many players and to employ them in their war roles (ie for mission or battle-staff rehearsal) then this would be OK. However, for a seminar game (particularly a strategic one) hands-on computer time would be a distraction from the important task of thinking and discussion.

(3) THE USER INTERFACE. For any game the "user-interface" is very important. Bluntly, is it easy to use?! This is especially important for computer-assisted games where the students have to get "hands-on". However, there is no point in the students having to waste time learning a computer interface which bears no relation to any they may have to use in reality. "Manual" interfaces such as the "tactical floor" or "operational wall" are discussed below at para 14b

(4) DATA STRUCTURES. The data structures used to store information are important and will affect the cost of maintaining the database. Is the data accessed through an easy-to-use, easily available database management system or is proprietary software used which needs special maintenance and training? In addition, is the data commented well or do the data files consist of strings of "meaningless" numbers? Is the data "normalised" and compact or scattered around a number of files, duplicating data and making it hard to update? All these points need to be considered..

EXAMPLE WARGAMES

9. There are many games currently in use world-wide. Examples range from theatre-wide "exercise drivers" to the highly detailed models used by operational analysts. A notable gaming organization is the Warrior Preparation Centre near Ramstein in Germany. This US facility uses a "confederation" of games (each of which is stand-alone) connected together by the Aggregate Level Simulation Protocol (ALSP) explained at para 15c below.

10. In the UK, the Royal Navy use a large number of games at HMS Dryad for training staff from 4* commanders downwards. Games used by the British Army range from IDAHEX/TAWS (used on the higher command and staff course at Camberley as an exercise driver) to Brigade and Battle group trainers. A tool called FLAMES (known in the USA as EADSIM or C^3ISIM) is used mostly for analysis of air defence, C^2 and air tactics. There are many other analysis tools such as ESAMS (produces SAM p_k), TAM (theatre attack model for aircraft/weapons optimization), SABSEL (produces air to ground p_k) and others too many to name here.

11. There are two main wargames in use in the RAF, the "TAC Thunder" wargame that is used on the Battle Staff Training Courses in the Air Warfare Centre and by the MOD Science Staffs and the "ACES" (Air Command Exercise System) wargame in use at the RAF Staff College at Bracknell. TAC Thunder is the subject of continuous update and improvement, as is the "ACES" game from Maxwell Air Force Base, USA. Comparing TAC Thunder and ACES illustrates perfectly the points raised above. Both games are very different and yet complementary:

a. TAC THUNDER. The game uses a simulation written in a computer language called "Simscript" produced by a firm called CACI in the USA, and runs on "Sun SPARC" computer equipment. The game is theatre-level, where a theatre can be as small as a state or as big as a continent. Scenarios can be generic or specific. The software does not restrict scenario selection and there is total control over the data without having to involve computer programmers to make the changes. TAC Thunder can be used for training or analysis, allowing either emphasis on true "reliability" of results or changes to produce "acceptable" results for training. The game works out the results using probability data and rules, although some element of determinism is possible by adding "orders" for commands, units and squadrons. Also, things like the timing and opening of air corridors can be set, and manual ATOs can be created, detailing targets and weapon configurations. The results that TAC Thunder produces are classified and have been validated by the US Studies and Analysis Agency in the Pentagon.

b. ACES. This game consists of two parts: a wargame "engine" written in Fortran (which only runs on "Cyber" mainframes) and a "front end" (which the user sees) consisting of a graphical map and a text based "tote board" for input, which runs on "Sun SPARC" or "X_windows" computer systems. The game can be theatre level or more tactical, the nature of the scenario being determined by the front end, a new version of which has to be written for each scenario. The current scenarios are optimised for training, though the "engine" is powerful and would be suitable for some analysis if a suitably "intelligent" front end was produced. The game produces its results by the use of embedded deterministic rule logic with some use of probabilities. Because of this, players must enter orders for every move an air or ground unit is to make. This can draw players "down into the weeds". Overall though, ACES is an excellent training game which is being vigorously improved.

12. This simple comparison illustrates quite clearly how the design of a game or analysis tool determines its suitability for different types of applications. The assumptions and simplifications embedded in the rules and data cannot be ignored, and so when comparing the suitability of games a thorough understanding of their strengths, limitations and methods of operation is essential. The glossy brochure view is not enough! The range of analysis tools and wargames available is, therefore, a vast and wide ranging resource.

SETTING UP A GAME

13. Setting up a game involves considering what is needed in addition to the game itself:

a. WHO WILL SUPPORT THE WARGAMES AND WHERE IS THE REQUIRED EXPERTISE? Some thought is required about who should support the wargaming. There is a shortage of wargaming professionals, especially for computer-assisted games. The formation of a suitable wargaming centre within your organisation could act as a focus for such support and assistance, where the wargamers would be centralised and so used most efficiently. Also, the wargaming centre could formalise the systems for exchange of experience and procedures between the current wargaming users.

b. WHO WILL TRAIN THE SEMINAR DIRECTORS? New wargaming seminar directors (SDs) need more than the standard "one week" handover normally used in the military. Wargaming SDs have specialist skills which are not in the main line of an officer's duties. Consequently, the handover should be during a wargame or analysis phase. So if there is only one wargame a year, then that is when the handover must be. If this is not done, the SDs will "muddle through" the next wargame, learning as they go, at the expense of the students/analysts. Of course, if there is a wargaming centre formed, then the SDs could go there at any time to acquire the appropriate skills.

c. WHAT EQUIPMENT IS AVAILABLE? Well established guidelines exist for selecting computer equipment, so this should be no problem. However, the issues of hardware compatibilities must be considered. It is also essential to establish what the peak load on the equipment may be. There is nothing that kills the credibility of a wargame faster than a slow and overloaded hardware system "cobbled" together from dissimilar elements.

d. WHERE WILL THE DATA COME FROM? Obtaining validated data is ostensibly not a problem. What will be a problem is getting it in the required format. Perversely, the most difficult data to get is information about NATO equipment performance because of commercial sensitivities. We really need access to our opponents "SECRET" books! However, when dealing with highly classified material, data exchange can become a nightmare. Which bit of data on a 500MByte disc is classified? Does anyone know? When was it last updated and from which source? Configuration control of data is essential and requires dedicated staff.

e. RULES. Rules embody doctrine, strategy, tactics and standard operating procedures (SOPs). Acquiring "correct" rules is vital to the success of the game. Rules though, especially once "hidden" in a computer, can have a perfidious effect on outcomes and their role should not be underestimated. Rules too need configuration control.

f. SCENARIOS. The development of credible scenarios is an art in itself. The scenarios must be consistent and "sensible" and they must be designed to meet and exercise the analysis or training problem.

g. WHO WILL MONITOR EFFECTIVENESS? Lastly, who is going to monitor the effectiveness and standards of the systems. If you are going to rely on the results of wargaming or trust the training of your leaders to these systems (including using them to assist in their decision making) then maybe you should know which systems you can trust and which you can't!

The final stage of the setting up procedure is the accreditation of the game. Who does this, and how it is done is not easy to solve. Issues of internal and external validation must be considered. Whether the game meets the objectives set is one issue. Just as importantly (and often ignored) is considering whether the objectives meet the requirements in the wider world. In wargaming, those wider world issues will literally mean the difference between life and death.

USING A GAME

14. Once a game is provided, using the game effectively (as has already been mentioned) involves further effort. How the game is to be used will affect decisions at this point.

a. THE ROLE OF THE STAFF. Whether the game is to be used for training or analysis, SDs, ie subject experts, are needed to interface with the users, be they students or analysts as follows:

(1) STUDENTS. SDs are obviously needed to interface between the game and the students, but how they do this is contentious. Many wargames are used as "exercise drivers" to stimulate interaction between opposing players. Here the wargame is merely an interface and the SDs act as umpires. In other situations the students "play" against the wargame and so its "behaviour" must be much more credible. The role of the SDs is now to act as the interface between the opponent (the computer) and the students. Note, however, that when evaluating the student's performance, the SDs will play a passive role by observing the group dynamics. In any case, unless the students need to get "hands-on" for a specific reason, there is little point in exposing them to the workings of the game and the full complexity of its results.

(2) ANALYSTS. SDs, ie subject experts, are also needed to co-operate with the analysts. Once a "question" has been posed, interpreting the question and the results produced is important. As an example, in an investigation of attacks on "strategic targets" the attacks were deemed to have little or no affect on the war at the "front". In fact, the attacks would probably have little effect other than to absorb air effort. The analyst concerned concluded

that attacking strategic targets was of no value (as it did not directly affect the battle) and therefore obtaining aircraft for this role was not a priority. This conclusion was entirely an artefact of the criteria used which judged success or failure by FLOT (forward-line-of-own-troops) movement and force levels. Subject experts can point out the problems with such a conclusion and prevent misunderstandings.

Thus, in both the above cases, management of the results is one of the most vital tasks. Even more important is what conclusions are drawn from the games. Interpretation is a tricky business and needs to be done with care, as is deciding on the criteria by which "success" or "failure" is to be judged. Certainly, as Desert Storm experience showed, better military training of (UK) analysts is required, so that "analyst's judgement" becomes "military judgement".

b. THE GAMING ROOM. In Naval wargaming the "Tactical Floor" (or a variation thereof) is often used. This represents an appropriate way to present the "game world" to the students/analysts. Considering what is the "appropriate way" to present the data from your wargame is vital. At the Air Warfare Centre we use the "Operational Wall" approach (see Figure 1 below), where, as the scenario is theatre (operational) level, a system for summarising the mass of data from our game is required. The Operational wall does this well. An essential part of the wargaming experience is for the students to concentrate on theatre-wide issues (with which they are unfamiliar) and ignore detail (which would be handled by staff lower down the system).

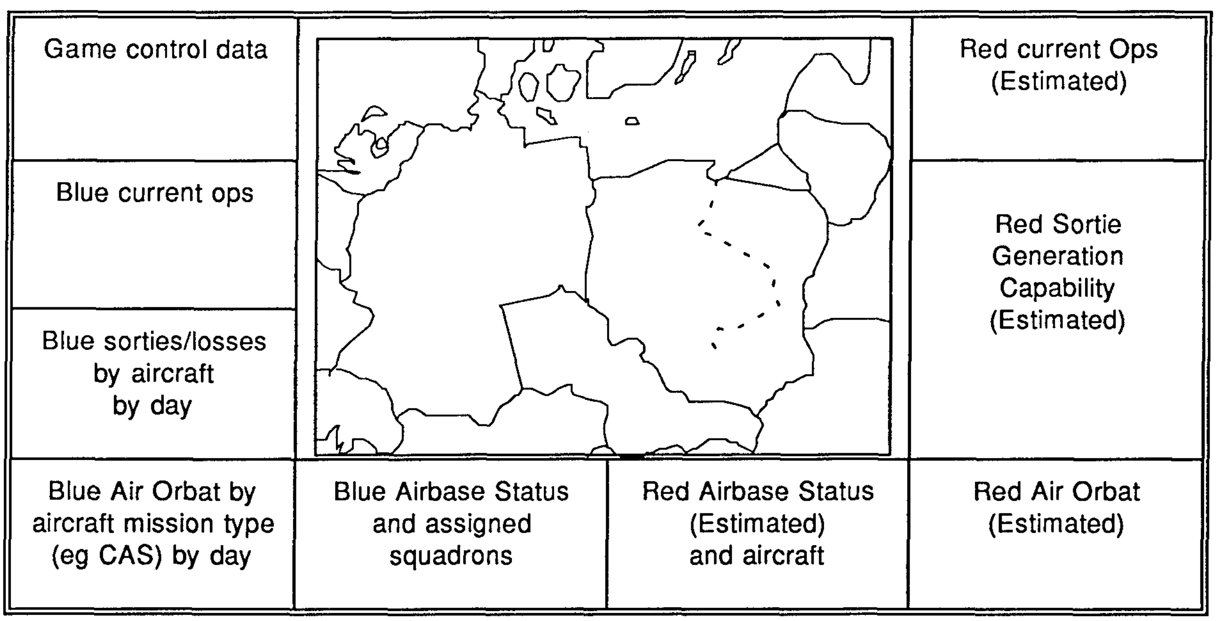


Figure 1 - Example "Operational Wall"

c. THE STAGES OF A GAME. Before a game starts good preparation is required. An appropriate scenario must be chosen and SDs briefed and trained in their roles. The SDs (umpires) are important in keeping the flow of information moving. The scope of the game in terms of geography, forces, command and control, team constitution and players must be decided. Also, the relationship of the game to real time (ie, is the game historical) and the "clock speed" must be set. The playing strategy and game objectives must be clear too. How "success" or "failure" is to be assessed (if at all) and what "measures of effectiveness (MOEs) are to be used, must be considered. The stages of the game are then as follows:

(1) PLANNING PHASE. The players must put on paper **their** strategic goals, estimate of the situation and appreciation of how the wargame may unfold. I would suggest the construction of a "campaign profile" using a "battlegram" (see Figure 2 below) to assist with this process. They must also set targets or criteria to be used to assess whether or not they are achieving their goals. For example: a goal such as "achieving air superiority" may be achieved when the enemy is no longer flying offensive sorties. This should be noted and the enemy's offensive sortie rate tracked.

(2) IMPLEMENTATION PHASE. In this phase the gamers have to apply and assess the progress of their plan, re-assessing and re-setting goals as necessary. It is crucial that the gamers "get inside" the enemy's "decision cycle" and take control. Learning to plan ahead and out-think the enemy's options is crucial to "success". This phase is essentially an iterative process and in wargaming occurs in the minds of the students, but in analysis has to be coded into some sort of analysis tool.

(3) THE DE-BRIEFING PHASE. The most decisive insights often occur during the de-briefing phase. The de-briefing (or results analysis) must be given adequate time and should be conducted skilfully by the SDs. The aim of this phase is to stand back and to detect the

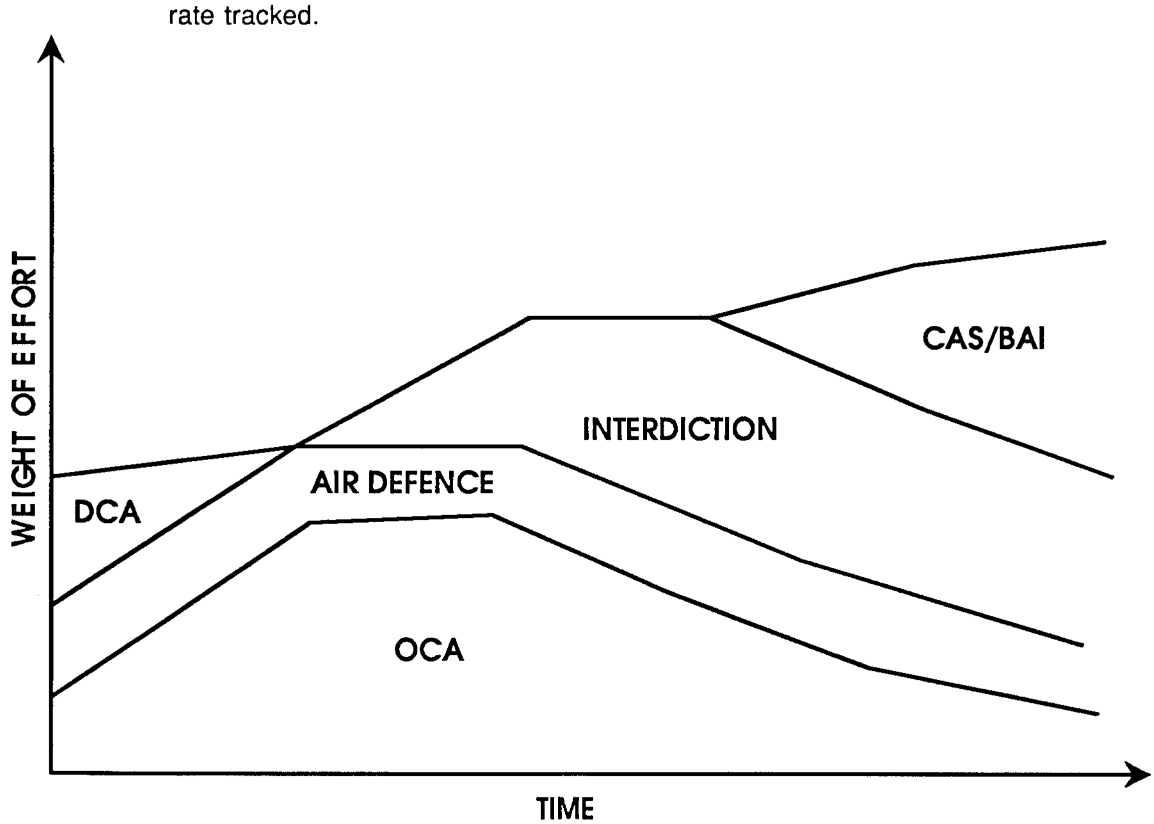


Figure 2 - Example Battlegram

patterns of events and analyze the interplay of friendly and enemy decisions. This phase may well force gamers to drop "reality assumptions" or "received wisdom" which they have, up to this moment, held to be true. An open-minded approach must be encouraged.

TECHNOLOGY ISSUES

15. There are new technologies on the scene which may well impact on future wargaming such as:

a. NEURAL NETWORKS. Neural networks are excellent at recognizing patterns such as the "common features in a large number of alternative military options" mentioned by Gavin Lidderdale (3) in his March 92 Air Clues article. However, while such techniques would enhance the performance of wargames, they would not make wargames a panacea for all our decision and analysis problems. Their role could be to be trained to recognise (and "advise" on) trends which are indicators of likely enemy action or of friendly crisis.

b. SYNTHETIC ENVIRONMENTS. The phrase "synthetic environments" has been coined to refer to systems of simulations and models connected across a network. Currently, only models at agreed levels of resolution can be connected, but once Tools For Aggregation and Dissaggregation (TFADS) are developed the potential is for high-level games to automatically draw their data from lower-level simulations or to be used to "drive" a confederation of lower-level models.

c. AGGREGATE LEVEL SIMULATION PROTOCOL. A notable gaming organization is the Warrior Preparation Centre near Ramstein (4) in Germany. This US facility uses a "confederation" of games (each of which is stand-alone) connected together by the Aggregate Level Simulation Protocol (ALSP). ALSP gets round the problem of having to produce monolithic models which do everything for every one by allowing a network of models to communicate. The models can be added/removed as required depending on the situation and so this mix-and-match approach gives flexibility. Consequently, the software maintenance task is smaller. ALSP allows models to exchange information about the entities they control, with other models entities appearing as "ghosts" in the "host" model. The protocol has to consider who has control over entities. To achieve this, each model has to have a bespoke "translator" which interfaces between it and the rest of the network. There is also a generic ALSP common module which passes the data out onto the net for consideration by other models and a generic ALSP broadcast emulator which manages network traffic, especially between "clusters" of models at a site. However, any new game has to conform to the existing ALSP protocol if it is to be allowed to join the"club".

d. VIRTUAL INTERFACES. A full discussion of virtual reality and its possible impact on wargaming is beyond the scope of this paper. However, it is worth noting my comments in para 14b above about providing "appropriate" data interfaces for students. Virtual interfaces for simulators? Yes. For wargames? Probably not.

e. SCENARIO COMPILERS. As has been mentioned, database "population" and maintenance is a considerable task. Moves to develop "scenario compilers" are underway. A scenario compiler collects and integrates data in the same way that a software compiler does. Across networks, a generic database formatting language is used to issue requests for information. Included in the request is information about how the data is to be formatted. This process needs the services of "agents" (see below) and would allow games to "self populate" and keep up to date.

f. "AGENTS". "Agents" are similar to Unix "Daemons" (and also to "viruses") in that they are self-sustaining pieces of software which can be used to "hunt" for information. They could use some of the "a-life" (5) methods to maintain their identity in the network. Agents could "work" for scenario compilers, asynchronously acquiring data and formatting it as required.

THE FUTURE OF WARGAMING

16. The future of wargaming seems assured. Even though the basic wargaming activity is likely to change little, I expect to see wargaming used in more diverse areas as follows:

a. FROM TRAINING MODE TO ACTION MODE. Over recent years "embedded training" systems have been developed which allow a piece of equipment to "toggle" between training mode and real use mode. In future, we may see wargames used in the same way, with pre-prepared likely scenarios being used to study possible conflicts and then switched into "reality mode" to act as the basis for decision making in crisis.

b. AS AN EXPERIENCE GATHERER. The developments in technology mentioned above in para 15 may allow wargames to accumulate "experience" which can be used to help predict outcomes. In any case, such "learning" would allow the games to suggest alternative courses of actions to students based on the game's "experience" of previous failed or weak strategies.

c. AS AN ADVISOR. Currently, decision makers may use wargames to assist their staffs with planning options, but it is possible to foresee a time when wargames have acquired enough "experience" to directly advise the decision makers on their courses of action. How such a tool would be "accountable" is an interesting point.

SUMMARY

17. Much money and time is spent using wargaming and analysis tools. However, the range available is vast, with little or no standardisation. As our use of wargaming increases, certainly within the RAF, there are considerable time and financial savings to be made if wargaming is effectively used and if the advice, training and support from wargaming professionals is co-ordinated and preferably centralised. The future uses of wargames in the military are still wide open. The recent establishment of the UK Defence Wargaming Policy Working Group (which aims to set standards and provide guidance on the selection, suitability and use of wargames tri-service) will help remove some of the uncertainty about the future of wargaming in the UK. However, it will be interesting to see how the other problems associated with wargaming will be tackled. Certainly our requirement for wargaming in the future is likely to increase, not decrease, as cost constraints on "live" exercises and trials continue. Nevertheless, the justification of wargaming on the grounds of cost savings or increased efficiency raises the spectre of how effectiveness is calculated. Funding for analysis has been extensive, but for training has been small by comparison. Its interesting to ask why. Finally, whatever task it is that you have in mind for wargaming, before you start, be aware of the mistakes made by others (and avoid them) and of the lessons learned (and consider them). As a last point, remember that:

"The analysis of outcomes is just the beginning of wisdom, not the end product". (6)

REFERENCES

1. The Pentagon. *Catalog of Wargaming and Military Simulation Models.* 11th Edition. Defense Technical Information Center, Alexandria, Virginia.

2. Alan J. Burton. *Computer Simulation (Some considerations on the relevance and the applicability of military modelling techniques).* MSc Thesis, Edinburgh University, 1992. Also

published by the RAF under the title: *Military Wargaming (An investigation into the Applicability and Validity of the Military use of Computer-assisted Wargames).* 1992.

3. Gavin Lidderdale. *The Analysis of Operation Granby.* Air Clues, Vol 46, N° 3, March 1992. UK MOD DT(F) RAF.

3. Alan J. Burton. *The Computer Wargame - Tool or Toy?* Air Clues, Vol 46, N° 4, April 1992.

4. Warrior Preparation Centre. *Computer-assisted Exercises and the Warrior Preparation Centre.* WPC, June 1991.

5. Steven Levy. Artificial Life - The Quest for a New Creation, Pantheon Books, New York 1992.

6. Peter P. Perla. *The Art of Wargaming.* Naval Institute Press 1990, Airlife Publishing Ltd.

7. *Various.* Simulation and Gaming Journal, University of Lille, France.

8. Peter J. McCarry. *This is not a Game (Wargaming for the Royal Australian Airforce).* RAAF Air Power Studies Centre, 1991.

9. Daniel B. Fox. *A Conceptual Design for a Model to meet the Wargaming needs of the United States Air Force.* Report N°: AU-ARI-84-8. Air University Press, Maxwell Air Force Base, Alabama, USA. July 1985.

10. Thomas A. Cardwell. *Wizard Warriors of Desert Storm.* Military Review, Vol LVXXII, September 1992, N° 9. Published by US Army Fort Levenworth.

11. John L. Krueger. *Pitfalls in Combat Simulations.* Military Review, Vol LVXXII, September 1992, N° 9. Published by US Army Fort Levenworth.

INFORMATION AGE COMMAND AND CONTROL TRAINING

Colonel Michael J. Swords and Jeff O'Byrne
Training Resources Management Division
Marine Corps Base, Camp Lejeune, North Carolina

Abstract

Until recently, command and control moved combat forces into position and watched them win the fight. New managerial techçniques made possible by broad band communications and the organization and storage of information in machine searchable data bases, have made command and control a battlefield of its own. These improved capabilities require major changes in the training required for commanders and their control agencies.

The commander's decision support system must provide appropriate information in time to support effective decision making. Decision makers must be able to maintain situational awareness from computer displayed information. Support for the training of these skills requires understanding of both the concept of command and control and the technological capabilities becoming available.

On the cutting edge of these events is the Marine Air Ground Task Force. Command and control of Marine Air Ground Task Forces must be exercised during the most difficult of warfare operations, an assault from the sea to enemy occupied territory. The command and control training for such a force therefore provides valid lessons for all services and a framework for the command and control of events in civilian contexts.

INFORMATION AGE COMMAND AND CONTROL TRAINING

Colonel Michael J. Swords
and Jeff O'Byrne

"Mindless warriors are to Third Wave War what unskilled laborers are to the Third Wave economy--an endangered species." (Toffler, 1993)

Command and control defines the human measures taken to organize and lead other humans in battle. It is a crucial area of modern warfare for two reasons: first, modern technologies make command and control a major force multiplier, that is the force with the most effective command and control will usually win the battle and certainly the campaign. Second, the mistakes of command and control are normally expensive and quite evident. Vincennes and Desert Storm fratricide can most productively be analyzed as failures of the command and control system.

Command and control has concerned warfighters for some time. As Van Creveld (1985) noted of Jena and Königgrätz, "If the Prussians triumphed in 1866 this was due to the fact that, like Napoleon sixty years before, they devised means to overcome the limitations of the new (command and control) instrument at the very time when they were exploiting its potentialities." Both had organized and trained their armies to use existing technologies to a higher standard than their opponents.

We have now participated in the first war of what Toffler calls the Third Wave Civilization but is more commonly the Information Age. As the Tofflers point out a new war form will "...profoundly upset existing military balance." (Toffler p179) In the Gulf War, victory and the continuation of regional military hegemony was determined by nationally controlled information resources. In the information age, such resources are readily available to all. Powerful personal computers, satellite imagery communication bandwidth, and Geographic Positioning Satellite (GPS) equipment are all freely available in the commercial market. The crucial difference in Information Age conflict will be the ability of a force to effectively use such readily available tools. This will be determined by training.

Such training requires a supporting structure fully as advanced as the command and control structure itself. It must operate inexpensively so as to allow repetitive training to mastery. It must be modifiable as quickly as the fighting system, but at significantly lesser cost. This paper will discuss the development of such a training system for Marine forces training at Marine Corps Base, Camp Lejeune and then identify elements required to fully support the needed training.

Doctrine provides the intellectual framework for military actions. Doctrine for United States services is published by the Joint Chiefs of Staff and the individual services. It states fundamental principles to guide the actions of commanders and units in combat. Education, generally provided in formal schools, is based on doctrine, and lays the broad, general foundation for combat decisions. Training in techniques and skills is provided so that commanders and staff officers can effectively use the tools provided.

DOCTRINE

Marine Corps command and control training is conducted per Fleet Marine Force Manual 0-1 Unit Training Management. (1988) which states, "The United States Marine Corps exists to keep the peace and, should war occur, to defeat the enemy.... The key to achieve this goal is training."

From this doctrine is derived a set of principles, Figure 1. They are useful in developing the tools required to support command and control training. Train as You Fight, the first principle dominates all of the others, for battle is the ultimate test of Marine Corps training. All peacetime training must conform to battlefield requirements. This means first, that command and control training must replicate the combat environment for commanders and their staff. Second, decisions and supporting staff coordination must have realistic outcomes. Third, commander and staff must have available during training, the doctrinal tools provided to develop those decisions and coordination measures.

The second principle recognizes that the commanding officer's leadership includes training his subordinates. Normally each headquarters trains subordinate echelon headquarters by directing and supervising training exercises.

The third principle recognizes the need for training to be guided by doctrine. While primarily used to guide FMF training, doctrine also provides guidance for the development of Tables of Organization (T/O) and Tables of Equipment (T/E). Training per doctrine therefore ensures that the unit is training with the personnel, weapons, and equipment with which they will fight.

Mission-Oriented Training requires that training programs be constrained to a properly conducted mission analysis. In the Marine Corps, results of mission area analyses are published for each type unit as Mission Performance Standards (MPS). Each commander may extend these standards by identifying a Mission Essential Task List (METL). From the METL, mission-oriented command and control training is planned. It is often conducted as a series of exercises within a single situation, with opposing force, terrain, and meteorological data all taken from real world references. After the first exercise is conducted, the exercise is halted and an after action review is accomplished. The planning cycle for the next operation is accomplished during the ensuing non-exercise period, with the troop lists and positions carried over to the second exercise.

Marine Corps doctrine satisfies the needs of the amphibious assault by establishing the Marine Air-Ground Task Force (MAGTF) as the basic fighting organization. The MAGTF comes in three sizes, (Figure 2), each with a command element, a ground combat element, an air combat element, and a combat service support element. The close integration of combined arms and maneuver under the MAGTF command element make the amphibious operation possible. It also requires that the command element be able to organize, task, integrate, and control these diverse elements. Command and control training includes normally attached units and routinely employs the full spectrum of their capabilities.

EDUCATION

Marine Corps officers are educated at the Marine Corps University, Quantico, VA and other service and joint schools. The purpose of professional education within the Marine Corps is to "...to develop creative, thinking leaders." (FMFM 1 Warfighting) The Marine Corps schools are tasked "...to provide a decision-making framework..." (Command and Staff 1994, p17). Marine officers start their education at The Basic School upon commissioning. The curriculum is continued as senior captains or majors at the Amphibious Warfare School, and as Lieutenant Colonels at the Command and Staff College. Upon graduation from these doctrinally referenced courses of instruction, the officer should be capable of executing the responsibilities incumbent upon their rank on a joint or service staff.

TRAINING

The MAGTF commander usually operates at the operational level of war while subordinate elements operate at the tactical level. This dictates different command and control training programs. Tactical unit commanders train in the shorter focus, more detailed control inherent in tactical operations. MAGTF commanders and staffs train to synchronize and integrate operations.

Training for Command and Control personnel must start with tool specific skills. This should be conducted immediately upon assignment to such a billet. Tool-based skills training is by no means a trivial exercise. The staff officer on a modern operational or tactical staff must be self-contained. The computer based tools provided are UNIX based, therefore a basic UNIX operator's course will be required. Although system specifications include X-Windows or similar graphical user interfaces, the importance of action officer trust in the machine requires a degree of computer literacy possible only with a good working knowledge of the operating system.

Each action officer must be capable of accomplishing the following:

System functions form the foundation upon which competence is based. Before starting a career as a modern staff officer the following functions must be internalized:
a. Maintain assigned memory devices so that loss of data is prevented, and retrieval can be logically and routinely accomplished.
b. Be able to operate system from the system prompt or with the front end provided.
c. Know system trouble shooting procedures for common file, hardware and LAN problems.

Since action officers are linked with each other and the headquarters information sources only by a Local Area Network (LAN), they must be proficient in its operation. Thus the staff officer must be able to:
a. Find, read, answer, and forward an E-Mail message from a co-worker on the LAN.
b. Find, read, answer,and forward an E-Mail message from stations outside the LAN.
c. Draft formatted messages and forward to releasing authority.

Within modern command and control systems, data is stored in a relational data base. The staff officer must understand the relational model and be able to use that knowledge to:
a. Know the data bases available within the system.
b. Know the type of data available on each data base, separately and in combination.
c. Find a specific list of data items in a data base by designing and formatting selection criteria.
d. Be able to sequence data items.

The Defense Mapping Agency(DMA) publishes geographical information in digital format for use in automated system. To the competent staff officer, use of these maps and associated Geographical Information Systems (GIS),

must be second nature. Thus the following functions are necessary:
a. Find a geographical location on a DMA digital map stored on CD-ROM
b. Zoom in and out on digital map without losing situational awareness.
c. Be able to conduct terrain analysis on digital map.

These tools are used to participate in plan development, therefore the staff officer must be able to:
a. Find, in an electronic library such as the Joint Electronic Library or KGB Fact Book an appropriate fact, extract, and place in document being drafted.
b. Draft and pass to another action officer electronically, a specific paragraph in a staff document such as an operations order, an estimate of supportability, or a fire support annex using application formatting tools to advantage.
c. Accept input from other action officers electronically and use to construct a single document.

Once tool specific skills are acquired, training leaves the mentor-student mode and moves to a guided problem solving mode. This provides the commander with the control needed to satisfy his responsibility for staff training as well as to facilitate learning the needed higher level skills. Because command and control training is only one type of training to be accomplished, it should be conducted as efficiently as possible, that is with minimal impact on the rest of the MAGTF.

Although it is quite possible to train a staff, by sending all elements of the force to the field there to await the results of staff deliberation, the effect on other training would be disastrous. A more efficient training method follows General Collins dictum that training is ineffective two echelons below the senior element [Collins p147]. This leads to the model for command and control training shown in Figure B. Note that the training headquarters is provided doctrinal communication channels with subordinate and higher headquarters. By accomplishing all exercise control measures through the higher and subordinate headquarters, the training population's perception of the situation replicates combat.

Depending on the exercise objectives, the Exercise Control Group might be a substantial organization or a single officer. Exercise realism determines the validity of the exercise. The realism is determined by the Turing Test,[1] that is, to the staff officers under training, the exercise should look like combat. Most importantly, their plans and coordination measures should be accurately reflected against possible enemy response. The Exercise Control Group must be adequately equipped to provide such service. Each action must be mediated in accordance with forces assigned, their condition, and performance in accordance with the best national intelligence and previous actions in the exercise and campaign. The intelligence and originality of the headquarters must not be limited by the training system.

The Marine Corps has three systems to provide tactical situations and resolution for command and control training. The first two are manual war games. TACWAR is a terrain board game, used to train squad and platoon leaders and company commanders. STEELTHRUST is map based and used to train battalion and regimental commanders and their staffs. The Tactical Warfare Simulation, Evaluation, and Analysis System (TWSEAS) is an automated tactical level war game. It will be replaced in 1995 by the MAGTF Tactical Warfare Simulation

[1] Postulated in 1949 by Alan Turing to determine if a computer could think. If written responses from a computer could not be distinguished from human responses, then the computer could think. (Hodges1983)

system (MTWS). These war game systems establish a tactical situation, resolve conflicts, and in the case of the automated systems, keep the books on the campaign, debiting losses and crediting reinforcements.

Efficient command and control training for tactical headquarters is conducted in a specially constructed facility, the Combined Arms Staff Trainer (CAST). Developed by Naval Air Warfare Center, Training Support Division, the CAST provides tactical staffs with housing and a communications system separate from the organic systems. CAST provides cells for the headquarters of seven infantry battalions, two regiments, artillery battalions and batteries, one division and the several functional agencies. All are linked by a hard wired audio system that replicates the doctrinal MAGTF communications.

Those who see the battlefield first hand, such as rifle company commanders and their supporting arms control teams, aviators, and reconnaissance teams, sit around a large horse shoe table to view the problem generator. The tactical situation is displayed by one of the war games or on large terrain boards within the horse shoe. The information flow within the CAST follows the model depicted in Figure B.

The final tactical test, after training with these systems and extensive field and live fire training, is the live Combined Arms Exercise (CAX) at the Marine Air Ground Combat Center, Twentynine Palms, CA. There the unit is put in a tactical situation, allowed to plan and execute fire and maneuver against target arrays. Direct fire is simulated by the MILES system, but indirect fire and close air support are live. The planning and coordination skills learned in the CAST are well tested in a CAX. This continuum produces well trained tactical staffs, capable of coordinating maneuver with fires.

No similar system exists for the operational commander and staff. Whereas the tactical commander is concerned with coordination of maneuver and close fires, the operational commander is concerned with sequencing tactical actions toward a strategic goal. The operational commander must coordinate the actions of land, air and often sea forces. “For the commander, the campaign is the basic tool to translate tactical actions into strategic results. In a campaign Marine leaders must...be able to integrate military operations with the other elements of national power in all types of conflict.” (FMFM 1-1 Campaigning, 1990 Foreword). Such a campaign is defined by the end state required to satisfy the strategic goal, and the sequence of operational objectives required to achieve the end state. The training objectives for such an effort are unique, and except for the initial tool-based skills, of a high cognitive content.

After a staff officer is educated and trained in system specific skills, three things remain to be accomplished before full participation in headquarters actions. First, the staff officer must become conversant with the routine staff planning and supervisory functions of the headquarters. Normally published in a formal document, these procedures establish the sequence of planning actions, and identify necessary points of contact.

The commander must impart to the staff, general guidance on how the organization will conduct the campaign henceforth. Horace Porter, tells of the conversations held by Grant's staff immediately prior to the Wilderness Campaign, “The eight senior members of the staff... participated in an intensely interesting discussion of the grand campaign...”. (Porter 1991)

After these preliminary steps the staff is ready for a series of gradually more difficult training exercises to develop higher cognitive skills. The difficulty is measured in two dimensions. First, the complexity of the situation can be varied to present more subtle indicators for the staff to evaluate. Second, time can be reduced. If, as Van Creveld claims, "...certainty is the product of time as well as of information, and the consequent willingness to do with less of the latter in order to save the former..."; then reasoned apportionment within the time-certainty formula must underlie all staff action. (Van Creveld 1985)

The training effort must focus on developing the higher level skills required to successfully prosecute a modern campaign. Inherent in war is the lack of perfect information, "...the history of command in war consists essentially of an endless quest for certainty...." but "...the attainment of certainty is, a priori, impossible.""(Van Creveld p 265) Therefore, the staff officer, as well as the commander, must, in a reasoned manner, balance certainty against uncertainty, always aware that time is a vital resource and that the cost of more perfect information is more time.

The staff officer accepts the guidance of the commander, develops plans to execute that guidance, presents them to the commander and supervises their execution. In Information Age warfare, this means that the situation presented on the information handling system must be analyzed, appended to the context, and opportunities and challenges identified. These opportunities and challenges must be examined for their impact on the ongoing operation, and various courses of action developed to further progress toward mission accomplishment.

The following discussion of the required intellectual activity is based on Adler and Van Doren's discussion of syntopical reading.[Adler and Van Doren, 1940] The synthesis of disparate information sources, required in this type of reading is very similar to the mental functioning required of a competent staff officer.

Initially, upon assuming the watch or commencing supervision of an aspect of the operation, the officer must inspect available information resources in order to find those relevant to the current situation. Each resource has an area of coverage, a specific reaction time and inherent resolution. The staff officer must be able to match the current situation with the most applicable resource. While this discussion is conducted in the framework of an action officer assuming the watch and therefore drafted in the present tense, during the planning phase, the staff officer must work the problem in reverse, i.e., identify the time frame and resolution to be required and then plan for the required resource.

In addition to differing reaction times and resolution, each of these information sources has a specific output format. The staff officer must therefore ensure that a common terminology applicable to all of the sources is available. This common terminology will normally be stated in terms of time and space. Force capability is measured by the ability to move through the battle space and project power. This raw data will normally be available, or developed, from joint and service doctrine, unit standard operating procedures, and the current commander's intent. Joint and service professional military education provides the basis for such a terminology. The organizational SOP expands it in detail to cover the specific procedures, and the commander provides the final layer by expounding his specific intent for this operation.

The next step is to frame a set of

questions to which all of the sources can be related. These may be formatted as templates, night notes, or pass down the line instructions to guide staff officers in evaluating the immediate situation. A template of capability might for example provide the attack indications for a specific enemy capability. These are developed from intelligence sources, modified by the current terrain and serve to call the staff officer's attention to specific indicators that predict certain enemy courses of action.

Once indicators have been assembled, they must be refined and presented to the commander for decision. The can most efficiently be accomplished by ranging the opposing indications from each source on either side of the issue. Figure 4 is the format provided staff officers of one organization for such presentation. By allowing for the weighting of each element, it facilitates more involved examination of the issues.

The staff officer must remember that an issue is not always defined explicitly between or among sources, but that it sometimes has to be constructed by interpretation of indications that may not have been their primary concern. It is necessary to analyze the discussion by ordering the questions and issues in such a way as to throw maximum light on the subject. More general issues should precede less general ones, and relations among issues should be clearly indicated.

A training facility to support Information Age Operational Level Command and Control Training is currently under construction at the Marine Corps Base, Camp Lejeune, North Carolina. Titled the Marine Component Command and Control Facility (MCCCF), it follows the information flow modeled in Figure 3. It will consist of two discrete areas, first, a command and control facility, connected with world-wide communications, as well as local tactical facilities, and equipped with a full suite of information processing equipment per the Joint Maritime Command Information System (JMCIS) specifications. Second, an Exercise Control and Debrief Facility will be provided to terminate the Time, Space, Position Information (TSPI) systems required to support complete debrief of DIS exercises. It will use the exercise control and feedback architecture, defined in the Institute of Electrical and Electronic Engineers (IEEE) Standard for DIS to prepare the signals for simultaneous, correlated display on any of several video displays. Space is provided for six action officers, nine flag, general, or commanding officers, and an audience of more than 200. Adding the MCCCF to the existing CAST and TWSEAS will provide a training complex capable of training tactical and operational staffs, either separately or together.

MISSING ELEMENTS

Four specific capabilities are still missing: backbone communications; DIS interfaces for existing simulators; instrumented maneuver areas and ranges; and Computer Generated Force (CGF) functional agencies.

It must be assumed that a joint task force will include elements from widely dispersed home stations. Marine Corps MAGTF's will, for example, be constituted of elements from Okinawa, California, and North Carolina, as well as at least one Marine Expeditionary Unit forward deployed aboard Navy amphibious shipping. To exercise such a force requires that replicas of doctrinal communications and exercise communications channels be provided to connect all elements from their home stations. This cannot be accomplished by use of normal communications means, since these are selected per the requirements of the objective area, not the home station location. For example, in the objective area, the regimental tactical net will be established as a Very High Frequency (VHF) net. Unit separations will match the 25 mile range of these tactical radios. In the exercise mode, however, the separations may be one hundred times that distance and will therefore require other means. The DIS standard provides for the digitizing, packetizing, and transmission of such signals as well as exercise TSPI signals. Per the standard, all communications will be packetized for transmission as Protocol Data Unit (PDU)'s. The backbone must provide sufficient bandwidth for these signals, as well as exercise specific requirements such as video teleconferencing for commander's briefings, exercise control and order wire during the exercise and after action review with the widely dispersed exercise participants. The Defense Information Systems Agency (DISA) is currently establishing a OC3 backbone entitled the Defense Simulation Internet (DSI). Bandwidth requirements are currently being estimated, and small scale exercises are being conducted to validate these estimates.

DIS interfaces for existing simulators and instrumented maneuver areas and ranges are required to support three mode ADS exercises. While, limited objective exercises can be conducted without full participation, the value of the man-in-the-loop makes such participation valuable. Fully implemented, Verification, Validation, and Accreditation (VV&A) for procurement actions will require such participation. The Marine Corps currently is operating or has under development 22 training and operational systems that should be integrated into the ADS environment. In the case of already fielded systems, this will require engineering of an interface to accept and generate, DIS compatible signals. In the case of most of the weapons trainers, it will require that their high level video be downgraded to the DIS standard. The tradeoff will be interaction with the targets on the screen. Nothing, however, presents overwhelming technical or funding challenges.

While an instrumented range is not needed for the DIS end state, they will be needed to solve communications and location problems in the interim. These ranges and maneuver areas must support unrestricted maneuver of combat vehicles, and report such dynamics per the DIS Standard into the DSI. This includes weapons flyouts and hit calculations.

The final missing element are Computer Generated Forces (CGF's) programmed to act as functional agencies. To date, the CGF effort has provided infantry figures that can run and chew gum at the same time. While useful, a more valuable function would be to provide important contextual tools for command and control training. To properly establish the context of an exercise requires the full functioning of agencies that cannot participate in the exercise. For some it is a matter of more important things to do. In a real world emergency, for example, TRANSCOM will have more important things to do than to execute the applicable Time-Phased Force and Deployment Data (TPFDD) plans. The phasing and sensitivity analysis of various portions are of critical concern to the JTF commander's tactical plans, therefore repeated runs of the simulation are required. A CGF, programmed with the applicable TPFDD elements, and able to generate various options within the plan could establish the arrival in country of JTF elements and hereby establish that parameter as part of the operational context. Similarly, national intelligence assets cannot participate in an ADS exercise because the physical enemy force laydown does not match exercise conditions. A CGF that included a model of applicable collection agencies and the enemy force laydown at various times, could provide valid execution of JTF collection plans and feedback to applicable staff officers.

CONCLUSION

The training of commander's and their staffs is crucially important to Information Age military organizations. Their ability to maintain situational awareness through a computer workstation, to orient the commander's intent to the perceived situation, and to draft clear, accurate orders to implement that intent determines the effectiveness of the force. Accomplished better and quicker than the enemy does, and success is facilitated. For the truth of the matter is that while privates and corporals can win a war, only captains and colonels can lose it (Griffith).

References

Adler, Mortimer J. & Van Doren, Charles (1940) How to Read a Book. New York: Simon and Schuster.

Collins, Lieutenant General Arthur S. Jr. U.S.Army (Ret) (1978) Common Sense Training. Novato CA: Presidio Press

Command and Staff College Philosophy. (1994) Student handout, Marine Corps Combat Development Center, Quantico, VA

FMFM 1 Warfighting. United States Marine Corps. Washington DC

FMFM 1-1 Campaigning. (1990) United States Marine Corps. Washington DC

FMFMRP 0-1 Unit Training Management. (1988) Marine Corps Combat Development Command, Quantico, VA

Hodges, Andrew (1983) The Enigma. New York: Simon and Schuster

IBM Dictionary of Computing. (1994) New York: McGraw-Hill.

JTP Pub 1-02 Department of Defense Dictionary of Military and Associated Terms. (1989) Washington DC: Joint Chiefs of Staff.

O'Byrne, Jeff (1993) Computer Generated Forces: Providing a Context for the Exercise Force. (1993) Proceedings of the Third Computer Generated Force Conference, Orlando, FL

Porter, Horace (1991) Campaigning With Grant. New York: Bantam Books

Toffler, Alvin & Toffler, Heidi. (1993) War and Anti-War:Survival at the Dawn of the 21st Century. Boston: Little, Brown and Company.

Van Creveld, Martin. (1985) Command in War. Cambridge MA: Harvard University Press.

Train as You Fight

Make Commanders the Primary Trainers

Train Using Appropriate Doctrine

Use Performance-Oriented Training

Use Mission-Oriented Training

Train to Fight and Support as a Combined Arms Marine Air-Ground Task Force (MAGTF) Team

Train to Sustain Proficiency

Train to Challenge

Figure 1- Principles of Marine Corps Training

Marine Expeditionary Force (MEF)

Ground Combat Element- One or more Marine Divisions

Air Combat Elementt- One or more Marine Aircraft Wings

Combat Service Support Element- Normally one or more Force Service Support Groups

Marine Expeditionary Brigade MEB)

Ground Combat Element- Regimental Landing Team

Air Combat Element- Composite Marine Aircraft Group

Combat Service Support Element- Brigade Service Support Group

Marine Expeditionary Unit (MEU)

Ground Combat Element- Battalion Landing Team (BLT)

Air Combat Element- Composite Helicopter Squadron

Combat Service Support Element- MEU Service Support Group

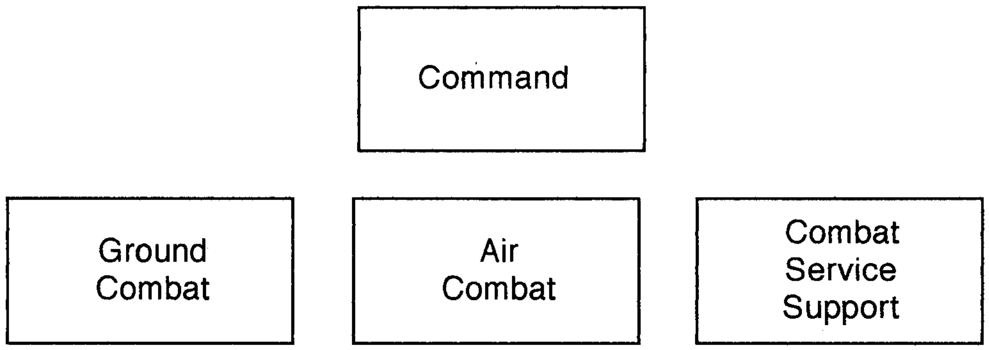


Figure 2- Marine Air-Ground Task Force (MAGTF). A MAGTF is composed of four elements: Command; Ground Combat; Air Combat; Combat Service Support

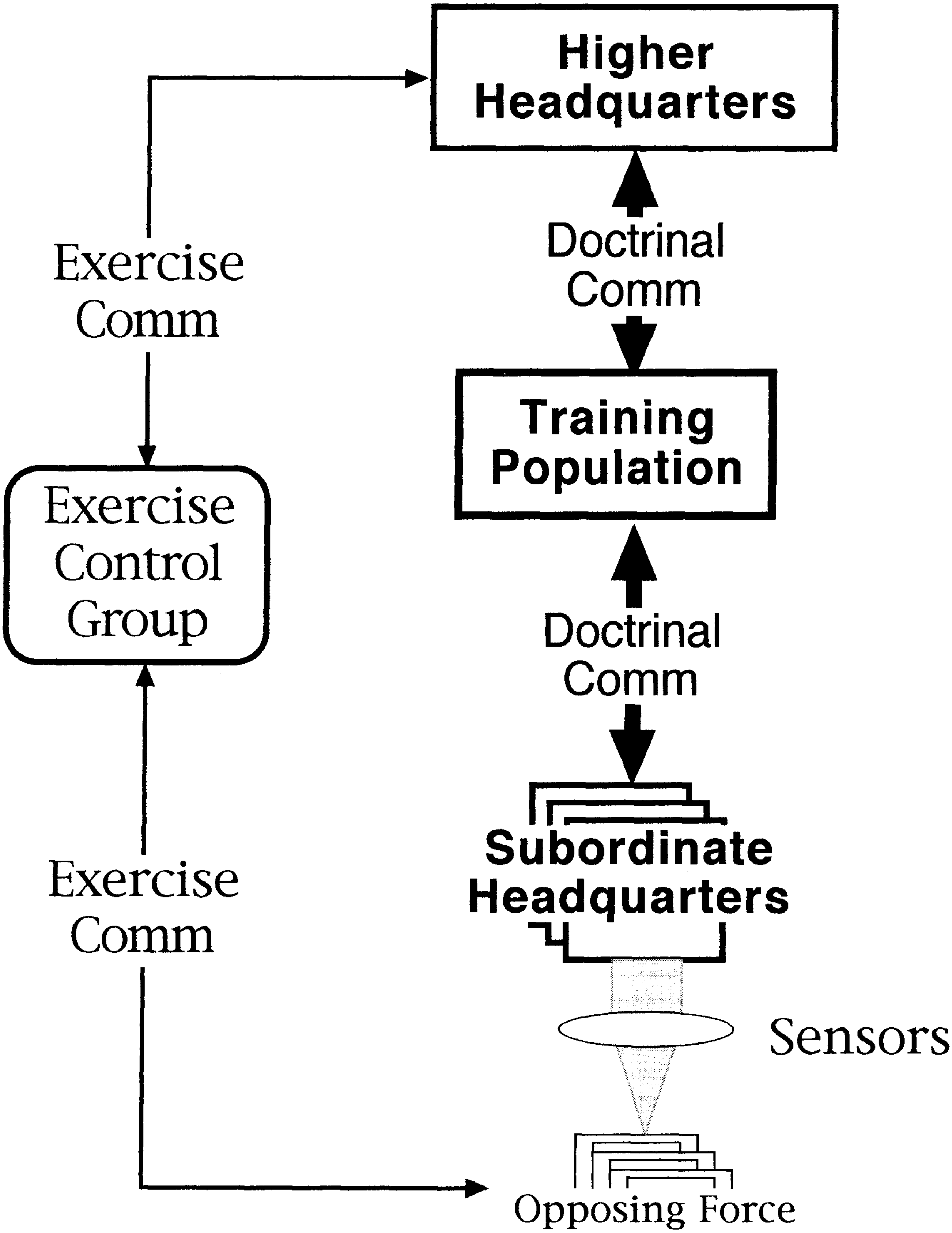


Figure 3- Model of information flow during training exercise.

GCE Analysis Matrix

Criteria	WT	CPA #1	COA #2	COA #3
Objective				
Maneuver Opportunities				
Mobility				
Tempo				
Force Protection				
Facilitate Future Opns				
Surprise				
Deception				
Logistics				
Command and Control				
Suitable				
Feasible				
Acceptable				
Comments				

Figure 4- Ground combat decision analysis matrix.

THE COST EFFECTIVENESS OF SYSTEMATICALLY DESIGNED TRAINING: LESSONS FROM THE FAA'S AQP PROGRAM

J.S. Bresee and A.G. Whitley
Delex Systems, Inc.
Vienna, Virginia

ABSTRACT

Instructional designers have claimed for years that systematically designed, outcome-oriented training is not only better but is also cheaper. As the argument runs, properly designed training addresses specific needs that have been accurately identified by prior analysis; only the training needed is developed and administered, thus saving wasted time and motion in the training of irrelevant or already acquired skills and knowledge. This argument has always had strong appeal, but has seldom been supported by data. This is probably partly because much systematically designed training is implemented for emerging systems where no basis of comparison with past training practices exists. Moreover, when systematic training design practices are used to revise training practices for existing systems, the effort is typically coupled with a major revision in content so that the effect of new training practices cannot be clearly distinguished from the effect of new training content.

The Advanced Qualification Program (AQP) for the initial and continuing qualification of commercial airline pilots offers a good opportunity for assessing the effects of systematic training design without the intervening effect of new and different training content. As an initiative to allow airlines to replace current training practices with an approach driven by well-documented analysis of training needs and requirements, the AQP will allow training professionals to observe the effect of new design on existing content in a well-bounded and well-understood domain.

This paper provides an overview of the AQP development process, and shows how an AQP provides opportunities for assessing the cost effectiveness of the products of instructional systems development. A training cost model is introduced as a potential dependent measure for use in assessing cost effectiveness and in predicting the costs/benefits of specific training design options. Preliminary results of the application of the model show support for the positive economic impact of systematically designed training.

BIOGRAPHICAL SKETCHES

J.S. Bresee has worked as a training analyst and designer for the past 17 years, with experience in training requirements analysis, training system design, and training capability development. He is employed by Delex Systems, Inc., where he is presently program manger for Advanced Qualification Program development.

A.G. Whitley has held leadership and management positions in technology-based businesses for over 15 years. He is the Training Division Manager for Delex Systems, and oversees all of Delex's operations related to Advanced Qualification Program contracting and technical development.

INTRODUCTION: ISSUES IN COST EFFECTIVENESS MEASUREMENT FOR TRAINING SYSTEMS

Since its inception, practitioners of Instructional Systems Development (ISD) have almost uniformly claimed that systematically designed instruction ought not only to be more relevant and appropriate to a specific training requirement, but also, by virtue of these properties, ought to cost less in the long run. Unfortunately, this long-standing claim of "better and cheaper" has resisted verification with a tenacity that has often seemed surprising. In retrospect, it ought not to have been surprising. Differences in content, training goals, and political environment have each had a diminishing effect on the comparison of relative training effectiveness. In spite of the many years of history in applying systematic instructional design, acceptance of its cost-effectiveness is still tends to be more a matter of faith than fact.

Differences in training content present an obvious obstacle to measuring training efficiency. Improvements in training practices have often been implemented along with improvements in the systems which trainees will operate or maintain. When this happens, the content of the new training program is often sufficiently different that any performance improvement resulting from new training methods is overshadowed, or at least obscured, by the improvements in the basic technology which the students are being trained to use. When, for example, systematic training accompanies the introduction of a new aircraft, improvements in human factors engineering may have resulted in systems that are significantly harder to understand, yet significantly easier to operate or interpret than the corresponding systems on the aircraft being replaced. Introduction of the Attitude/Direction Indicator (ADI) responding to a central air data computer as a replacement for a gyro horizon and gyrocompass presented this sort of situation. In other cases, the systems of the new aircraft were both more capable and more difficult to learn. In aviation, this occurred when the flight management system (FMS) replaced the autopilot and VOR/TACAN for enroute aircraft control. In either case, the content addressed by systematically designed training is sufficiently different from the content addressed by more traditional training that direct comparison of relative training effectiveness and associated cost is difficult.

Another obstacle to direct comparison has been changing training goals. When the goal of an existing training course is described as the "understanding" of a system, while the goal of a systematically designed replacement training course is stated as the detection and correction of system malfunctions, the increased specificity of the desired outcome is probably sufficient to yield a very different set of training and testing strategies. When students are asked to "gain an understanding" of a system, not only is the learning goal non-specific, the model of instruction that is in place is generally that of the public school classroom. This sort of goal might be appropriate when a broad subject area (say, for example, English Literature) is under consideration, the availability of teachers is limited, and testing can only occur by means of sampling the domain.

Generally in these cases, the teacher "surveys" the content, students do extensive additional reading, and tests which sample a domain of knowledge are administered periodically. This sort of approach requires to some extent that tests remain a mystery. When the complete domain of content is so large that it can only be surveyed, and when tests can only sample this domain, prior knowledge of the test would allow a student to learn only the tested sample of the domain, and invalidate the instructional intent of the course.

Of course, just the opposite is true in most technical training. Here, specific tasks comprise the training content. The typical "survey" objective borrowed from academia is not as appropriate under these conditions. Still, this sort of objective shows up in some aviation training systems, especially as part of aircraft system overview training, Here, the stated learning goal (as distinguished from a behavioral training objective) is often something unobservable and unmeasurable like "gain an understanding of the function of the hydraulic system." Goals of this sort are found especially in older or smaller commercial aviation training curricula, and instruction developed to support such a goal will differ significantly from training designed to enable the performance of more specific objectives.

Finally, in some cases, much time and effort has been spent in justifying and implementing what was then a new and very different training development process, often requiring much persuasion to pave the way. In this situation, it would not be surprising if a program manager elected not to emphasize formal summative evaluation of the program; the career consequences of ambiguous or negative results could be severe.

In summary, then, the evaluation of the effectiveness of systematic training has suffered from significant differences in the basic technical content, significant differences in the specific goals of the training curricula being compared, and lack of incentive to take the measurements. What is needed to make the case for cost effectiveness is an environment where newly designed training can replace an earlier training system addressing the same content and oriented toward the same student evaluation criteria.

This situation can presently be found in commercial aviation where airlines are re-designing crew training systems under the FAA's Advanced Qualification Program (AQP). This initiative allows those airlines who undertake a complete analysis of their operational tasks to replace current training and evaluation practices with new events which are tailored to each airline's own operational requirements. This initiative provides a nearly perfect laboratory for the demonstration of the cost effectiveness of systematically designed instruction. Most importantly, the core content both before and after the implementation of new training is exactly the same. The aircraft to be operated has not changed, nor have the manufacturer's published procedures, the airline's operating practices, or the route structure in which the crew must fly.

Under an AQP, an airline establishes a new set of qualification standards that take on the force of regulation for its crew members. Based on systematic job analysis, these qualification standards reflect airline-specific training and evaluation needs for the equipment, route structure and crew population of the airline. The governing special federal aviation regulation (SFAR 58) and accompanying advisory circular (AC 120-54) require that the new training and evaluation system and standards be at least as stringent as current practices, with the objective of improving levels of crew qualification above the present standards as provided in FAR parts 121 and 135. In practice, this has meant that aircraft handling standards have remained as presently published in existing practical test guides, but an increased emphasis on crew resource management and line-oriented settings for training and evaluation has been added. These additions have caused some changes in training and testing strategy, but it can be argued that they are the result of systematic instructional development, and are therefore part of the instructional treatment. In summary, methods and techniques of training and evaluation have changed, but every significant aspect of the job itself, including standards of performance, has remained the same. Clearly, the AQP application represents a nearly ideal ISD laboratory.

The political or programmatic reasons for avoiding comparative evaluation of training effectiveness is eliminated by FAA direction. AQP participants are required under the governing SFAR to submit evaluation performance data with student identification removed on a periodic basis to the FAA branch overseeing AQP development and implementation. This requirement acts to ensure a valid basis of performance comparison between training approaches.

AQP PROGRAM DESCRIPTION

An airline develops an Advanced Qualification Program by following an explicit Instructional Systems Development (ISD) process with analysis and documentation requirements that fully meet the intent of MIL-STD-1379D. While many of this standard's internal documentation requirements of questionable usefulness are eliminated, the overall program requirements will be familiar to any ISD practitioner. The airline must explicitly identify the behavioral components of the job to be performed, and establish qualification standards based on specific observable behaviors. A supporting curriculum outline must be developed which provides proficiency-based training in support of the approved qualification standards. Evaluation of each qualification standard based on a terminal proficiency objective takes place in a line-oriented (mission-oriented) environment, either during a line check or an annual line-oriented evaluation (LOE) session in an advanced flight simulator. The result is a replacement of "one-size-fits-all" airline training with specific training solutions for individual airline requirements.

An AQP is typically developed according to the advisory circular in a five-phase process. These phases are:

Phase I:	Initial Application
Phase II:	General Curriculum Development
Phase III	Training System Implementation and Courseware Development and Implementation
Phase IV	Initial Operations
Phase V:	Continuing Operations

Beginning with the formal application, an airline carries out explicit, documented analysis, design, implementation and ongoing evaluation of the resulting training system. Supporting documentation requirements can be met conventionally, or through more innovative approaches that are individually approved. For example, when airline operating documents are concise and readily accessible, a comprehensive set of references to these documents serves very well to document supporting skills and knowledge. In such cases, analysis requirements can be met with a comprehensive task listing, a complete set of qualification standards, and references to specific pages of operational documents.

Once training and evaluation requirements are established through analysis, a curriculum is designed that allows for proficiency-based assessment of student performance throughout training. This design process is followed by courseware and training media development and procurement where necessary. Initial implementation takes place in Phase IV, with ongoing program operation and evaluation carried out in Phase V. The process builds in the flexibility to reduce training costs through innovative media choices, tailored curriculum outlines, and other specific modifications that allow adjustments to each airline's individual needs.

MEASURING THE RESULTS: A MODELED APPROACH

There is no question that airlines have a unique opportunity to improve the quality of their training programs and significantly reduce training costs under the AQP initiative. From the very outset, it was apparent that there was a need to develop a clear understanding of the changes in training curricula, media, and training costs for airlines considering a transition to AQP. There are three primary analytical challenges that drive this need. First, a detailed understanding is essential for evaluating the impact of specific changes to existing training programs as well as comparing the cost-effectiveness of alternative training solutions. Therefore, it is necessary to identify, measure, and relate the utility and costs associated with new training approaches in a systematic fashion. Second, evaluation of AQP alternatives requires consideration of a number of variables and cost factors that must be addressed for each airline as a separate entity. Each airline develops and maintains its own training program with unique characteristics that will dictate training scope and associated cost magnitude. Third, implementation of AQP must be monitored over time to track the actual versus estimated costs of training programs that are developed.

In order to meet the analytical challenges of AQP development, specific tools were required to assist with analysis during implementation as well as for subsequent tracking of AQP results. The approach taken was to develop an AQP cost model to be used for three primary purposes: (1) to calculate initial estimates of cost savings resulting from implementation of an AQP; (2) to conduct comparative cost analyses of alternative AQP implementation plans; and (3) to document initial estimates of cost savings for subsequent verification and validation by actual costs savings once an AQP was implemented. A tailored AQP implementation and tracking system to document airline cost savings from systematically designed instructional programs under AQP as well as the expenses associated with AQP implementation was also developed. The data captured in this latter system form the basis for subsequent verification and validation efforts of initial AQP model predictions.

The two applications comprising this cost forecasting and tracking system provide highly accurate data on net airline savings from transitioning to AQP. In effect, the cost model and accompanying cost tracking software provide a closed-loop system for predicting, assessing, and verifying cost savings resulting from systematically directed changes to airline training programs. The AQP model and tracking system were developed using Borland's Paradox relational data base management system, providing a relatively straightforward, economical and portable set of tools for analyzing training applications.

The AQP cost model is the essential tool for estimating the cost of training operations. It is logically structured through a data base architecture which partitions the input data for the airline and cost parameters into five fundamental modules:

> An Airline Equipment Module for basic airline data inputs including aircraft fleet size, number of aircraft by type, and aircrew ratios for each aircraft type over a five-year projection period. This module is required to structure basic elements of the AQP analysis.
>
> A Labor Cost Module for the cost data for training instructors and aircrews (Captain, First Officer, Flight Engineer) for each aircraft type. The module factors wages, taxes and fringe benefits and includes inputs for travel and per diem costs for aircrews associated with training sessions. It establishes crucial cost parameters for the AQP analysis.
>
> A Training Resource Module documents current and projected training costs for simulator time, flight training devices (FTDs), ground school instruction and any specialized training that is required for each aircraft type.
>
> A Curriculum Module for data on utilization rates for each of the training resources required under existing programs or projected with a new curriculum after AQP implementation. This data is also unique for each aircraft type in the fleet.
>
> An AQP Implementation Module documents alternative costs associated with analysis and curriculum design tasks, procurement of training media, and the costs of sustaining training following AQP implementation. This module is essential for establishing all costs associated with implementation of AQP for each airline.

Based upon the parameters established within each of these data base modules, cost-benefit calculations reflecting alternative training approaches and resulting cost savings are provided by the model.

The primary value of the AQP model is in assessing the feasibility of new training approaches by calculating potential cost savings resulting from their application. The AQP cost model is specifically tailored to address current commercial airline aircrew training requirements including initial, transition and upgrade training, and the corresponding training courses developed as part of an AQP. Costs associated with existing training programs are compared with forecasted training costs following implementation of AQP. Forecasted costs include initial costs for establishing and implementing an AQP program within an airline, and the ongoing costs required to maintain the program. In a typical application, a five-year projection of costs based on existing training programs will be compared with training costs forecasted following implementation of AQP training. The five-year projection period provides a useful context for incorporating consideration of future changes the airline plans to make, such as changes in fleet size or aircraft types, and assessing the cumulative benefits from AQP training over time.

The model is capable of addressing a variety of training options and reflecting cost implications in great detail. It is structured to include all important variables associated with existing and future airlines training options, and is therefore capable of being rapidly reconfigured to conduct sensitivity analysis and update cost estimates.

RESULTS AND CONCLUSIONS

The airline training cost model has proven to be a valuable tool in supporting the selection among the training design alternatives that AQP development presents. In developing and verifying the model, training analysts obtained cost information for existing training programs from several major and regional airlines, and found that the model's predicted costs were in generally close agreement. In most cases, this agreement was rapidly achieved, with only small adjustments of variable values required to achieve accurate cost predictions. In those cases where significant errors in prediction were found, further research led to a modification of the model to account for the specific circumstances that were operating in these special cases. Accurate predictions of current costs suggest that the forecast of savings under AQP are based on a solid foundation.

The AQP cost model has been applied to more than 15 airlines for purposes of predicting costs savings resulting from AQP training. At the present time, tracking data have only been collected over a significant period of time for one regional airline during the early stages of AQP implementation. While this sample size is certainly not adequate to draw strong conclusions, the results to date provide solid support for the predictive value of the model. In this example, actual net savings (AQP savings less costs of implementation) have been realized to within 2% of model predictions over seven months of operations.

The close correspondence between forecast and actual cost savings observed to date tends to lead to two specific conclusions about the model itself. Firstly, the current model does in fact capture the important cost variables associated with airline training operations. Secondly, the accurate prediction of cost savings lends support for the structure of the model with respect to the relationship of these variables. It seems reasonable, therefore, to conclude that a cost model of this basic design is a useful tool in management assessment of training cost savings due to changes in training operations. When combined with performance data on qualification standard behaviors, the basic data is available for an assessment of training cost efficiency.

Use of the cost model has led to a more general conclusion that airlines pursuing AQP conversion of their training systems are demonstrating real cost reductions as a result of systematically designed training. By moving training and checking closer toward actual line operations and away from contrived training maneuvers is an clear application of the military aphorism that "you fight like you train." In all cases studied thus far, this has led to more cost efficient training as well, with savings in the first year which are often sufficient to fund AQP program development efforts.

Modeling current practices serves as the basis for prediction. Developing a cost model that can accurately predict current costs from observably-changing readily obtainable facts is reasonable insurance that such a model will accurately predict the effects of changes in training practices that affect the component variables of the model. This approach would seem to be applicable to any training system and its associated costs as both a program planning tool, and as a dependent measure of its overall effectiveness.

THE COMBINED ARMS TACTICAL TRAINER FOR THE BRITISH ARMY

Roger Burch
Brian Rush
Procurement Executive, Ministry of Defence
London, United Kingdom

ABSTRACT

Changes to the political situation and threat in Europe together with a greater public awareness of environmental conservation have resulted in pressure to reduce live military training. Additionally, the British Army is facing other training constraints due to cost, safety and range availability. Against this background is the need to maintain operational effectiveness and any substantial shortfall in field training will need to be made good in other ways.

The Army Strategy for Simulation in Training has stressed the priority that must be given to systems which compensate for the lack of field training resources by allowing basic skills and work-up training to be completed in barracks. The core of the Army's simulation programme is to be the Combined Arms Tactical Trainer (CATT) that will allow approximately two hundred armoured vehicle and helicopter simulators to be networked together in a realistic combat scenario. The CATT must allow all battlefield assets to be deployed and fully integrated in two-sided exercises from platoon to battlegroup level.

A number of Pre-Feasibility Studies into CATT were conducted in 1992/3 and five Feasibility Study contracts were let during 1993 and reported in mid 1994. These show there are obvious comparisons to be drawn between the CATT and the very similar US Army's Close Combat Tactical Trainer (CCTT). Their respective In-Service dates are also virtually coincident. However, a number of significant differences have been identified, which are discussed.

The methods used by the Procurement Executive of the MoD vary from US methods, and these are explained. The procurement options available to MoD are also highlighted and discussed, together with specific areas of risk as perceived by PE.

THE AUTHORS

Roger Burch has worked in the Programme Directorate/Military Command Control and Information Systems since 1992 as a senior project engineer in the UK CATT Project, which was responsible for selecting, controlling, and assessing the CATT Feasibility studies. He joined the Ministry of Defence, National Gas Turbine Establishment in 1967, and graduated in Mechanical Engineering at City University, London in 1972. He has worked in procurement in a number of Departments including Guided Weapons Production policy, and as a Resident Project Officer at British Aerospace, Hatfield, responsible for all Air-to-Air Missiles. He has also worked in the contracts area providing technical costing and risk assessment advice, and working on bid assessments.

Brian Rush joined Programme Directorate / Military Command Control and Information Systems in 1993 as project engineer for the CATT project. He is a representative of MoD within the UK Simulation Interoperability Working Group and Distributed Interactive Simulation Workshops. He joined the Ministry of Defence Science and Engineering Fast Stream in 1990 after graduating in Engineering from Cambridge University.

THE COMBINED ARMS TACTICAL TRAINER FOR THE BRITISH ARMY

Roger Burch
Brian Rush
Procurement Executive, Ministry of Defence
London, United Kingdom

INTRODUCTION

The Combined Arms Tactical Trainer (CATT) is an interactive simulation system intended to provide in-barracks training of tactics, combat drills and procedures for armoured vehicle crews from platoon to battlegroup level. A battlegroup has no fixed composition but typically will comprise 2 tank squadrons and 2 armoured infantry companies together with associated battlefield assets such as reconnaissance, forward artillery observers, anti-tank sections and engineers. It is commanded by a lieutenant colonel and can be compared with a battalion in the US Army.

Aim of paper

The aim of this paper is to provide an overview of the UK procurement process, an insight into the CATT programme and the current status of the project, and to explain how this fits into the overall training strategy.

Contents of paper

The paper will address the following topics :

- Introduction
- Concept for CATT
- Functions of the Procurement Executive
- A comparison of CATT with other training systems
- Interoperability and DIS
- Preliminary views of the UK Feasibility Studies
- Conclusions

CONCEPT

The Concept for a Combined Arms Tactical Trainer was originally drawn up by the Director General Army Training (DGAT) who is responsible for defining the Army's training strategy.

Severe constraints are being made at present on the use of field exercises and traditional ways of providing training in combat skills. These constraints include escalating costs, safety concerns, environmental issues and a decline in the availability of training areas. Although in recent times, the UK has made extensive use of a number of training areas in Germany, access to these is now being increasingly restricted.

It is a part of the Army's strategy to redress this shortfall by providing in-barracks training through the medium of computer simulation. This avenue has become technically achievable only in the past decade, but systems such as SIMNET have already demonstrated that a viable alternative to field training can be provided.

Additional benefits that may accrue from an increased use of simulation include improved exercise assessment and after action review, greater training transfer and the ability to conduct exercises which would be too hazardous to perform in the field.

PROCUREMENT EXECUTIVE

The role of the Procurement Executive (PE) is to procure equipment for the UK Armed Forces. It is broadly equivalent to the Army Material Command, although it takes responsibility across the board for Army, Navy and Air Force equipment procurement. It is useful to understand the relationship between PE and the Armed Services and to see how this can affect a project such as CATT.

Background

Staffed by a combination of civilians and serving military officers, the PE organisation emerged during the late 60's and early 70's with the aim of improving procurement practices and ensuring value for money. The role of PE is to take the Armed Services requirements and convert them into a specification that can be understood by Industry, and against which the eventual equipment can be delivered.

In common with much of the Government service, the PE has undergone a number of organisational changes over the years aimed at improving its efficiency and effectiveness. The first milestone in the process was the Downey Report published in 1966, which introduced a phased procurement process with distinct breaks and a system of formal reviews throughout a project lifecycle. This was followed by the Rayner Report in 1971 which introduced the concept of project teams, with all of the procurement functions controlled by a single Project Manager. This was the point at which the title of Procurement Executive was first used.

Later in the 80's came the increased favouring of competition throughout the Government Service. The final significant changes were brought about by a report conducted under the guidance of the Prime Minister's Efficiency Unit in 1988 which made a number of recommendations in the light of experience gained on various high-value projects. These included an increased focus on risk management, a preference for commercial-off-the-shelf procurement and encouragement of performance demonstrations as early as possible in a project life cycle, by the development of prototypes and technology demonstrators.

As a result of these changes, there is a greater concentration and integration within our Project Management process, and this applies across all projects.

The Project Manager

The Project Manager is responsible for all procurement aspects of the project from "cradle-to-grave". This in practice means from receipt of a Staff Target from the user, one of the armed forces; through development and production; logistic support while in service; to final disposal. To assist him he has a small project team dedicated to the overall control and day to day running of the project, together with a number of functional staff that are an integral part of the team. These include Finance, Quality and Contracts specialists who work directly under the Project manager's control.

In addition there are organisations within the Procurement Executive from which the Project Manager can draw upon specialist advice and assistance. These include Integrated Logistic Support, Reliability and Maintainability, MANPRINT, Risk Assessment and Life Cycle Costing.

Current government initiatives favouring the increased use of commercial practices within the public sector have resulted in the formation of 'agencies' responsible for their own budgets and potentially competing with industry for some aspects of government work.

An example is the Defence Research Agency (DRA), formed from the former defence research & development establishments. A considerable quantity of scientific research and technological investigation is conducted by the DRA on behalf of the PE each year. Whilst formerly funded independently from within the defence budget, the DRA now utilises internal charging for work conducted on our behalf. If the pace of current reforms continues, this may become the format for much of the project office's interactions with other bodies within government in the future.

Project Plans

A project will typically follow a number of distinct phases (see figure 1).

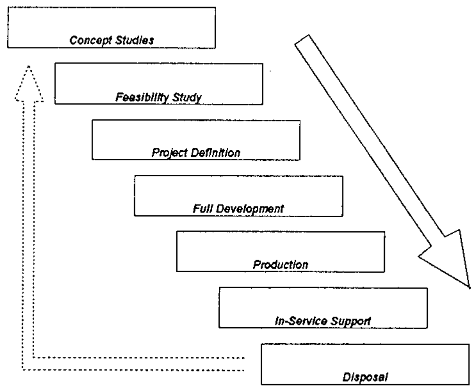


Figure 1 : The Downey Cycle

Concept Studies may be undertaken by the relevant service branch, and in the case of CATT these were co-ordinated by the Land Systems Operational Requirements (LSOR) division in the central staffs of MOD. For training systems, the Director General Army Training (DGAT) will usually also assist. Such studies are generally conducted in-house, but may call upon the expertise of the Defence Research Agency or of external consultants. In our case it was DGAT who drew up the initial Concept Paper.

From these studies the Operational Requirements branch acquired sufficient understanding to issue a Staff Target, against which Feasibility Studies could be conducted. These studies, which reported in May of this year, were intended to assess the feasibility in technical, programme and cost terms of meeting the Army's needs. The studies were placed as a result of a world-wide competition, conducted against a specification drawn up by the project office. The findings of the studies are currently being analysed in order to arrive at a procurement strategy which meets the customers' needs without presenting too great a risk to successful completion. Simultaneously, the Operational Requirements branch are drafting a firm 'Staff Requirement' which will become the prime document defining the system to be procured.

This is the point at which the CATT project currently stands. The next stage is a crucial one, and must be carefully planned in order to ensure best value for money for the user whilst containing risk within manageable limits.

Traditionally, the next stage would be Project Definition - aimed at defining the system configuration and investigating further specific risk areas identified by the Feasibility Studies. This would be completed prior to a commitment to full development and production.

With SIMNET having demonstrated the concept of networked simulation, and to some extent the effectiveness of training so derived, and with CCTT at an advanced stage of development we see little overall technical risk to the CATT program. That is not to say that there are not significant risk areas, the selection of an image generator and the assessment of training performance being examples, but that we perceive these risks as quite manageable.

There are many factors which will have a bearing on how we may proceed, not least of which is our contracting policy. For example, the risks which we currently perceive might best be mitigated by means of a spiral development similar to that of CCTT. However, UK government policy strongly favours firm priced contracts with clearly pre-specified deliverables, factors which do not lend themselves well to a spiral approach.

The options that are available to us include :

- Conduct a further paper study.
- Place a limited development contract to include the manufacture of hardware and demonstration of performance in key risk areas.
- Proceed straight into full development.
- Consider joining a comparable programme being undertaken by another nation.

The first option, conducting further studies offers a low expenditure route for the next stage but is unlikely to provide us with a great deal of further information not uncovered by the feasibility studies. Prior experience has shown this to be ineffective at reducing risk and that this is likely to result in a considerable delay to the in-service date.

The second option, a limited prototype development, is more attractive as it provides a real opportunity to experiment with a variety of configurations - in particular assessing visual system performance prior to making a final selection. This would inject additional costs and delays into the programme, however, especially if, as is usually the preference, two contracts were to be run in competition.

The third option, that of proceeding straight into full development, is clearly also possible as much of the hardware and software we require is available now off-the-shelf. It might reduce the procurement timescale, but would expose us to risk which may cause any initial savings in time and money to be lost.

The fourth option, to consider joining a comparable overseas programme, would have the advantage of significantly reducing technical and program risk, but may potentially have an impact on the cost and flexibility of the final delivered system.

Whatever the route we eventually take, it is unlikely that an Invitation to Tender (ITT) for the next phase will be issued before mid 1995, or a contract placed until the end of that year.

COMPARISON OF CATT WITH OTHER TRAINING SYSTEMS

UK Trainers

The UK has adopted NATO classification for its Army training systems, designating them as Level 1, 2 or 3.

- Level 1 systems are skills trainers specific to a particular vehicle, and examples include gunnery and driver trainers for a main battle tank.
- Level 2 trainers provide tactical training to crews of a variety of vehicle types. This category includes tactical engagement simulation as well as CATT-like systems.
- Level 3, command and staff trainers provide training to the higher command levels at brigade and above.

CATT, as a level 2 trainer, must simulate the function and performance of individual vehicles, but only to a fidelity sufficient to perform tactical training. Certainly, replication of the fidelity and functionality of vehicle-specific level 1 trainers would be costly duplication. However, we must equally be careful in designing a "selective" fidelity trainer that is sufficiently faithful to the original equipment so as not to incur negative training.

An example of this need to maintain a fine balance, is the training of precision gunnery. CATT is not intended to teach the finer points of tank gunnery skills, indeed there is a level 1 trainer for that purpose. However, if a gunner does not experience a realistic response and kill a target within the simulation when he reasonably expects to, then this will detract from his appreciation of the overall training experience. Thus we have introduced the term "accurate gunnery" to differentiate between precision gunnery requirements and the objectives of CATT.

Other Nation's Trainers

Because CATT is required to fit within a broader training strategy, the number of other nations assets to which it can be compared, is somewhat limited. SIMNET and CCTT are clearly similar systems, and the German Leopard 2 platoon trainer and Marder trainer (known as AGPT & AGPG respectively) have much in common with CATT. The feasibility study contractors were tasked with assessing each of these systems with a view to identifying their critical features and, in particular, their suitability for adaptation to form the CATT system.

Looking first at the German trainers :

An AGPT system comprises three networked simulators and a control station, each housed in its own portable container. It uses SIMNET-like communication protocols and a GT 120 Image Generator of late 1980's vintage. Although a useful and effective trainer for the purpose for which it was intended, we feel that it is unlikely to be capable of expansion to the kind of system we envisage.

The Marder trainer, AGPG, is a much larger and more complex simulator comprising up to 24 networked simulators arranged in groups of 3. DIS protocols are used, and it has excellent visual imagery and high physical fidelity. Although it has the potential to meet our requirements, the system has features over and above our current aspirations and, for the numbers we would require, is likely to exceed our available budget.

The audience will be familiar with the US systems SIMNET and CCTT. SIMNET has shown the training benefits of networked simulators, but is now considered rather dated technology. We are doubtful that the same system could be cost effectively manufactured today. CCTT is the system most closely aligned with our need and has received a great deal of attention during the course of our studies.

The principal differences between the two systems are listed in table 1.

	CCTT	CATT
Scale	Company group of manned simulator modules at each of many sites	Battlegroup of manned modules at each of 2 sites
Visual System	Fixed 15 Hz update rate throughout	Minimum 24Hz update rate required for accurate gunnery.
Simulated Vehicles	Simulated vehicles include M1A1/A2, M2/M3, M119, etc.	Challenger 2, Warrior and Scimitar are the principal vehicles
Computer Generated Forces	Friendly forces utilise US doctrine	UK doctrine required for friendly forces
Dismounted Infantry	Control provided for individual section commanders of dismounted infantry	Control unlikely to be required below platoon commander level
Wide Area Networking	No current requirement for Wide Area Networking	WAN required to connect UK and Germany sites
Interoperability	Interoperability with SIMNET a requirement of the initial modules	Not required to interoperate with SIMNET, but possibly with other UK training systems.
Mobile Units	Vehicle mounted mobile units required for National Guard training	No requirement for mobile units, although transportability may be required
Terrain Databases	Generic European and Desert area databases required	Salisbury plain and BATUS training areas plus NW Europe
Training Philosophy	Methods of training and requirements for performance assessment and after action review are likely to differ significantly	
Procurement Strategy	Spiral development with cost-plus development contract	Firm price contract favoured

Table 1 : Comparison of CCTT and CATT

It can be seen that even with a system that closely resembles CATT, any collaborative arrangements would not be straight forward. There would always be a number of nation specific differences, although with careful planning these should not be insurmountable.

INTEROPERABILITY

The objective of CATT is to train units of up to battlegroup size in military tactics, combat drills and procedures. This can be achieved using homogeneous, networked simulators located on a single site. DIS provides the potential to interoperate with other simulators which could potentially be anywhere in the world. It has been hailed as the advent of 'seamless simulation'.

Following the initial euphoria, however, interoperability does not appear quite as easy or indeed as practical as was first envisaged. It is early days yet, but link-ups such as those we have seen at I/ITSEC and elsewhere have demonstrated some of the limitations of heterogeneous simulator interconnection. There are likely to be serious terrain and image generator correlation problems for sometime to come whenever there is a need to link heterogeneous systems. We question the training value of such link ups and note that CCTT at present does not intend to incorporate long haul networking.

There appears to be no pressing military need within the UK to link CATT with other systems, and although the Army has aspirations for further interconnections, there is little evidence of firm requirements at present.

By way of illustration, observe the Artillery Fire Control Trainer (AFCT) project. This is a system, comparable with the US GUARDFIST 2 programme which is due in service in the UK in 2-3 years time.

The requirement is for a group of three networked simulators, able to manoeuvre around a simulated battlefield, and intended to provide training for Forward Observation Officers (FOO) in tactics and procedures, in working together and directing artillery fire. It is likely to require DIS compatibility.

On the surface, it would appear to be an ideal candidate for interoperation with the CATT system and could yield potential cost savings if as a result the artillery observer element of CATT could be omitted. Problems are being realised, however, which make this increasingly unlikely :

- Budgetary constraints may limit the Image Generator performance of AFCT, and we would not wish to constrain CATT performance to the lowest common denominator.

- The terrain database procured for AFCT is likely to be of a different size and fidelity standard from those required for CATT.

- AFCT will be heavily utilised for concentrated procedural training of artillery observers and will have little spare capacity for CATT exercises.

Thus, for a number of technical and administrative reasons highlighted above we are unlikely to be pursuing interoperability between CATT and AFCT, inspite of very clear potential benefits.

In the long term we do see opportunities for interoperability between UK simulation systems, but these are likely to evolve gradually as technical difficulties are overcome and clear requirements emerge.

Interoperation between training systems of different nations is an even more difficult area. Intellectually it would be satisfying to connect CCTT and CATT, or to link with some of our European partners. However, in practice, units from different nations rarely have any operational need to communicate or interact significantly at battlegroup level or below.

Inspite of the problems which have been highlighted, it is likely that the UK will gradually introduce greater interoperability and adoption of the DIS standards but we acknowledge that the benefits may not be as great or as immediate as initial reactions may have suggested.

PRELIMINARY VIEWS FROM THE UK FEASIBILITY STUDIES

In May of this year the final reports from the 2 main feasibility studies conducted by Logica and Link-Miles/EASAMS were received. Three subsidiary studies were also conducted into specific technical risk areas : Interoperability, Terrain Databases and Modelling, and Visual Imagery. These reported in February and their findings were fed into the main studies. The project office has spent much of the Summer assessing all five studies and considering the way ahead.

The studies began with a detailed Training Needs Analysis which provided a focus against which technical assessments could be conducted. There then followed a detailed technology investigation including a comparison of systems both planned and in service with the UK Army and the armed forces of other countries.

Considerable attention was devoted to MANPRINT and Integrated Logistic Support activities and finally a number of procurement options were presented with associated costs and timescales.

The main recommendations from the reports were as follows:

Representation of Battlefield Players

The vehicle modules in which the trainees would participate in a CATT exercise should be represented in one of the following ways :

- High fidelity modules - closely replicating the crew stations of the principal vehicles, namely Challenger 2, Warrior and Scimitar, and having representative sights and controls.

- Generic simulators using high resolution visual displays but with non-representative controls, and a generic module enclosure. The same type of generic module could be used for engineering vehicles, personnel carriers and helicopters.

- Workstations with simple controls a plan view display and communications equipment for logistics and command functions.

Accommodation

The simulators and complete CATT system should be housed in a large commercial warehouse type building or brick structure, one in UK and one in Germany. Some vehicle modules should be containerised in order that elements of the CATT system can be transported for deployment at short notice.

Interoperability

The studies recommend compliance with the latest version of DIS. We see DIS, currently at version 2.0.4, as satisfying 95% of our interoperability requirements. The UK Simulation Interoperability Working Group (SIWG) actively promotes UK interests at the DIS Workshops in Orlando and we are confident that the standards will serve CATT well.

Visual System

The studies naturally identify the visual system as a key element within CATT, and the selection of the correct image generator as a critical decision for the project office. It is necessary to clearly identify our requirements in order to meet the need fully without over-specification. However, the ability of the visual system to provide a range of features such as changing atmospheric conditions, battlefield smoke, realistic range detection levels and dynamic terrain without noticeably impairing the visual performance is equally important. Image Generator technology is advancing rapidly and simultaneously costs are reducing. We feel that it is important not to commit too early to a specific visual system, in order to get the best performance possible for the money we have available.

Terrain Database

The terrain databases to be used in CATT do not necessarily need to represent a specific geographical area in detail, as the objective is to train tactics in a variety of terrain types and scenarios. Accordingly, generic or even wholly synthetic databases might suit our requirement. However, in order to facilitate progressive training from CATT through Tactical Engagement Simulation (TES) and Live Field Training Exercises (FTX), we would wish to have databases of Salisbury Plain in the UK, and the BATUS training range in Canada.

The acquisition of raw terrain data is not seen as particularly difficult, nor the software tools for processing that data into a runtime database. However, the successful integration and optimisation of the database with the chosen visual system will be a significant task, with the scale of databases required being in excess of 100km square.

Computer Generated Forces

CGF is perceived as posing potentially the highest technical risk for CATT and represents probably the greatest area in need of development work. It is known that a number of CGF systems exist, but on investigation one finds that many are solely developmental or operational analysis tools not intended for training applications. The range of suitable off-the-shelf products is at present quite small.

Our requirement is for control by one operator of forces up to battlegroup in size. The studies have indicated that realistic control at platoon level is readily available, and that company level control will be available in a year or so. Realistic and effective computer generated forces at battlegroup level is still, we feel, some way from being achieved.

Clearly, we need to consider the best approach for a system which is due in service by the turn of the century.

One option might be to not set our sights too high initially and to adopt a low risk approach, accepting what is proven, tried and tested at company level. Battlegroup level CGF could be introduced as a mid-life update when the technology is more mature. Even this is not entirely without risk - currently available software tends to originate from the US and would need modification to represent the behaviour and tactics of our own forces. Nor would it be without cost, with a requirement for up to 5 battlegroups of supporting and opposing CGF, the additional workstations and CGF controllers required in the interim would add considerably to the price of the system.

After Action Review

The ability to conduct performance assessment and after action review will contribute greatly to the training effectiveness of CATT. A formal performance assessment system will require a resident military group in order that consistency is maintained between the training of different battlegroups.

Technical risk is involved as the system will be capable of gathering huge quantities of digital data, such that data processing will become a major task. With so much going on in a CATT exercise, it will be near impossible for the assessment group to monitor all activities. Rather, they will be dependent upon some form of automatic processing and labelling of key events and learning points from each exercise.

Systems already exist that can generate statistical outputs such as numbers of rounds fired, time to kill, etc. Our requirement is likely to go considerably further than this including such aspects as missed firing opportunities, poor tactical use of terrain, and so on.

Integrated Logistic Support (ILS) and MANPRINT

Our studies have provided advice on ILS and management aspects such as system Availability, Reliability and Maintainability (ARM), MANPRINT, Risk Assessment and Investment Appraisal. These will assist us in writing the specification for CATT and assessing company bids against value for money criteria.

CONCLUSIONS

The Combined Arms Tactical Trainer for the British Army is a system with, on the face of it, a lot of similarity with the US CCTT program. It is of similar technical complexity, intended to fulfil a related training need and is being procured according to a comparable timescale.

The UK feasibility studies have identified a number of differences between the training objectives that will affect the performance of the two systems and the way in which they will be operated. In particular, the studies have indicated the need for a higher fidelity visual system, a higher level of control of Computer Generated Forces, and more sophisticated performance assessment software for after action review.

There are fundamental differences between the US and UK methods of procurement, the most significant being the UK's current competition policy. Fixed price contract conditions are favoured to minimise the risk exposure of the UK Government.

The procurement options available to the CATT project office have been discussed, ranging from a "conventional" UK approach as explained, to joining forces with a comparable programme undertaken by another country. Whatever the route we eventually take, we feel sure that the project will generate much interest within the simulation community in the coming years.

INTEGRATING USERS INTO SYSTEM DEVELOPMENT: USER EXERCISES IN CCTT

Thomas W. Mastaglio, PhD and Everett A. Goodwin III
CCTT Integrated Development Team, Loral Federal Systems
Orlando, FL 32826

ABSTRACT

CCTT is a networked training simulation system being developed for the U.S. Army STRICOM through a series of seven incremental builds. These builds will progressively add system components and increase the complexity of components delivered in previous builds. Each build will integrate the previously built system with newly delivered hardware and software components into a system which is partially functional. Total system functionality incrementally increases until at the conclusion of build seven the system is complete and can enter qualification testing. To increase assurance that system testing will be successful and that CCTT is ultimately training effective, a user assessment of each incremental build is conducted. These assessments are conducted in the context of operational user exercise scenarios with Army users. Each scenario is designed to train those collective tasks which can be performed using the technical capabilities provided by the system functionality in the CCTT system built thus far in the program. The user exercises provide both a checkpoint on progress toward meeting the technical requirements of the CCTT program and a way to assess the system's training effectiveness. Training effectiveness is assessed based upon collective tasks which are going to be evaluated for training transfer during the system's initial operational test and evaluation (IOT&E). The approach supports a continuous test and evaluation philosophy while gauging the training effectiveness of a system throughout its development. The methodology used in CCTT is key to integrating a user focus into a concurrently engineered training system being incrementally developed.

ABOUT THE AUTHORS

Thomas W. Mastaglio is the Loral Federal Systems CCTT Training Effectiveness Advocate. He retired from the U.S. Army in 1991. His final military assignment was as a training developer at Headquarters Training and Doctrine Command. Dr. Mastaglio earned a Ph.D. in Computer/Cognitive Science from the University of Colorado in 1990.

Everett A. Goodwin III has been a Federal Systems employee for 15 years. Prior to his appointment as the IDT Manager he was the Training Manager for Federal Systems Integration at the Federal Systems Company Manassas site. Mr. Goodwin is a graduate of Florida Atlantic University in Ocean Engineering.

INTEGRATING USERS INTO SYSTEM DEVELOPMENT: USER EXERCISES IN CCTT

Thomas W. Mastaglio, PhD and Everett A. Goodwin III
CCTT Integrated Development Team, Loral Federal Systems
Orlando, FL 32826

OVERVIEW OF CCTT

The Close Combat Tactical Trainer (CCTT) is a Distributed Interactive Simulation (DIS) Standards compliant system for training Army Company/Teams and Platoons in the collective tasks required by their unit type Mission Training Plans (MTP) [1]. The system requirements and conceptual design for CCTT were derived from the ARPA SIMNET technology demonstration [2, 3]. CCTT will be fielded at fixed sites starting in 1997 to the Armor and Infantry Schools and to posts which garrison Armor or Mechanized Infantry Divisions in the United States, Germany, and Korea.

Mobile versions of CCTT are being designed to train Armor and Mechanized Infantry platoons. Current plans are for these platoon sets to support distributed training in National Guard combat units. CCTT is comprised of a network of simulators. Simulator modules represent the primary weapons systems and support vehicles found in the close combat portion of the battlefield. These include, the M1A1 and M1A2 Abrams Tank, the M2A2 and M3A2 Bradley Fighting Vehicle, the HMMWV, the Fire Support Team Vehicle (FIST-V), and the M113A3 Armored Personnel Carrier.

The Army needs CCTT to fill critical training deficiencies resulting from more restrictive access to training areas, funding limitations, and increases in the sophistication of weapons technology [4]. Units will train in CCTT under unit control. It will have the capability to simulate Battalion Staff tactical functions by emulating real world equipment such as the SINCGARS radio interacting with combat crews in their simulators.

Computer workstations provide a way to interact with computer generated forces which perform battlefield functions in the virtually simulated world (e.g., combat engineer units conducting counter-mobility operations). Opposing forces (OPFOR) and friendly units on the battlefield located near the training unit will be emulated by Semi-Automated Forces (SAF) [5].

INTEGRATED SYSTEM DEVELOPMENT

CCTT is presently under development by an Integrated Development Team (IDT) comprised of engineers from six companies and the government [6].

The IDT has organized its development effort into seven incremental system builds. These builds are designed to integrate, throughout the course of the development phase, the CCTT functionality which has been developed to that point by concurrent engineering teams responsible for each of the system components. This spiral system development methodology is conceptually illustrated in Figure 1.

The incremental builds both add functionality (i.e., components) to the maturing system and increase the capabilities of each component. For example, the first CCTT build integrated an M1A1 simulator module, an After Action Review Component able to record and replay the training exercise, a DIS network, and the system Master Control Console. The second build increased the capabilities of that M1A1 module, incorporated the capability to process remote entities, and integrated a rudimentary Semi Automated Forces (SAF) component.

The idea behind the incremental system builds is to integrate a complex system step-wise throughout development, allowing engineers to identify early any inconsistencies or problems that impact proper integration. The availability of a partially functional CCTT system offers a unique opportunity to assess its progress toward achieving the objective of providing effective training for the soldiers who are the ultimate system users.

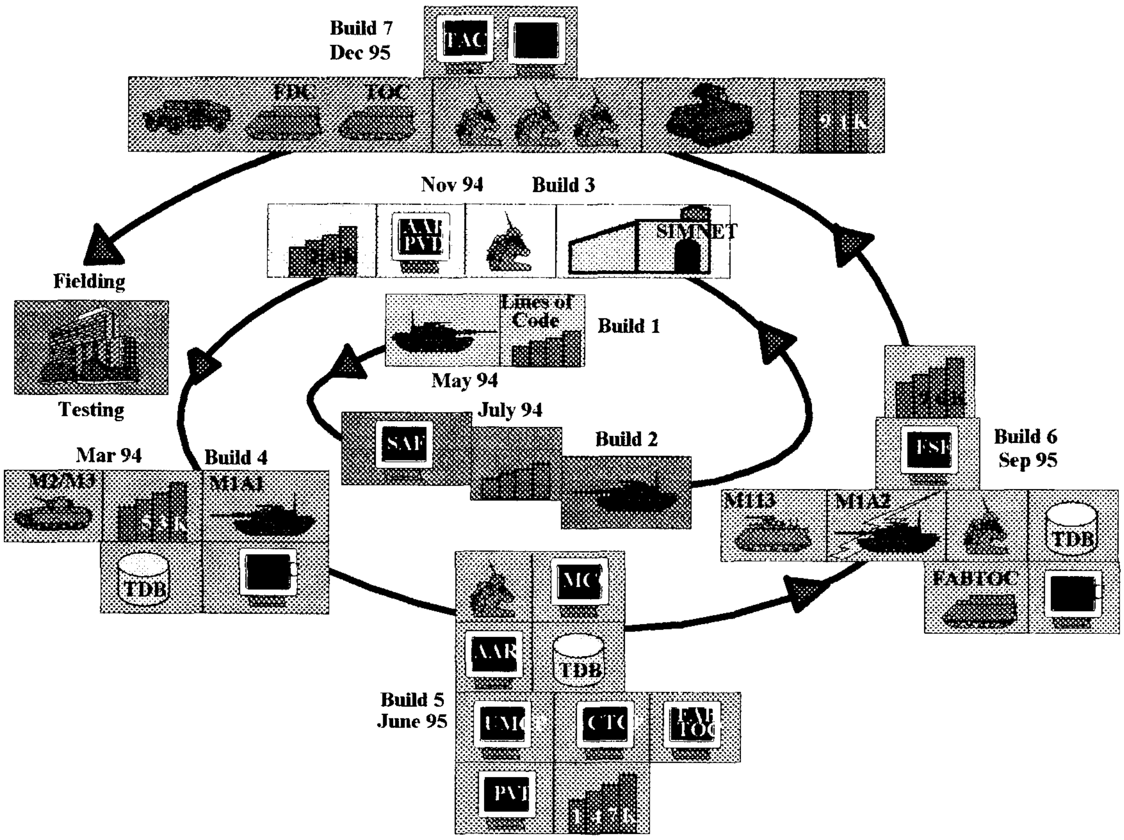


FIGURE 1

RATIONALE FOR EVALUATIONS

By examining the functionality available in each build and cross walking it with collective tasks to be trained in CCTT, we develop a user exercise that can be performed at the conclusion of each integrated CCTT build. This gives us an opportunity to conduct early system level evaluations against system operational requirements as well as checking technical specifications. Our approach is not necessarily unique. For example, the Air Force Training Systems SPO developed an approach to system testing called Simulator Test 2000 [7] that makes extensive use of subject matter experts as members of a Critical Process Team. Their focus is evaluating training products and services while CCTT is the first full scale development of an Army collective training system.

CCTT is the largest and most complex training system the Army has ever procured, and in terms of the number of its components, it may have that distinction for all of the military services. Basing CCTT user exercises on crew and collective tasks integrates an operationally oriented evaluation of the system into the development process sooner than it would be using just traditional formal operational testing approaches used for final system checkout.

There are numerous benefits to this approach:

- Engineers get to see soldiers using their system; they get direct feedback early enough so that it can effect design and development for subsequent builds.
- The user community sees the contractor's progress toward meeting their requirements, and
- Users are brought into the development process early, rather than being handed a

system at operational test and then asked to decide if it meets their needs.

The approach is not without its drawbacks, significant attitudinal changes are required:

- The user community has to be level set to understand that the system is immature and that the early builds do not completely resemble the finished product. Users must be convinced that they are part of the team developing the system and that their input is needed to assess what has been done to date.
- Industry engineers need to be assured that they will not have to "start over" after every evaluation because a new set of requirements has been generated.
- The military procurement community has to accept a major paradigmatic change. They must be comfortable with allowing their customer (the system user) to see how industry is progressing; they have to control user expectations.
- A process for managing user feedback, specifically providing it in a useful format to engineering, must be developed.

Strong industry and procurement community leadership is required to make theuser exercise concept effective and accepted.

USER EXERCISE APPROACH

To develop the user exercise approach for CCTT we conducted a front end analysis of collective tasks that could be supported using the technical capabilities of each build. For each build we develop tactical scenarios based on those tasks, design a data collection methodology, conduct the exercise, and finally analyze the raw data gathered from user feedback to determine a set of technical issues.

To accomplish all of these steps a working group was formed consisting of development engineering staff, human factors engineers, system integration staff, training and doctrine experts, subject matter experts, test engineering, and software engineering. This working group is co-chaired by the industry Training Effectiveness Advocate and the Assistant Army Program Manager. This group authors, reviews and approves the training scenarios for each build, the plan for conducting the user exercises, and the final report of results.

To determine which tasks would be included in a user exercise, the team starts with the technical capabilities provided in a CCTT build, that is, the system build to be evaluated. This set of capabilities is cross walked with a previously completed analysis of the tasks that can be trained in CCTT. The latter are documented in a database called CATTASK built by a government support contractor. Electronic access to CATTASK facilitates rapid information retrieval during the analysis.

A subjective assessment is made of which tasks can probably be supported. This task set gets refined as scenarios based on them are developed and tested. The scenarios are not scripted chains of events, but rather a set of tactical conditions to which the training unit must respond that are set in a situation training exercises (STX). The actual STXs used are modifications, as necessary, of those defined by the Army in their Mission Training Plans for Armor and Mechanized Infantry units.

The scenarios are tested during system integration and adapted where needed. In some cases, we find that not all predicted collective tasks can be trained in the next build. We then refine our analysis to show which ones would in fact be evaluated within a given scenario. As integration progresses and in preparing for the user exercise the scenarios are "dry run" by personnel assigned to the IDT. These personnel include the subject matter experts who are part of the Army Optimization Team [6], active duty soldiers assigned full time to support the program on site.

The actual user exercise (USEX) is conducted by a 15 person team drawn from the Working Group. Integration and Test personnel operate the system assisted by development engineers from each CE team. The purpose of including development engineers is so that they can view first hand how their system component performs and observe user interaction with it. This has proven useful in helping the CE teams understand technical issues arising from user comments. Responsibility for directing the evaluation is shared by the CCTT Training Effectiveness Advocate and the Assistant Program Manager from the Army responsible for test and evaluation. Control of specific scenarios is managed by a government subcontractor who authored the scenarios.

Two types of users participate. Army uniformed personnel operate the simulators and also observe the exercise in a trainer or observer/controller role. In the latter role, they guide the after action review process. Contractor personnel fill the same roles that Contract Logistic Support (CLS) personnel will perform at an actual CCTT site. The roles are system operator, SAF commander, and AAR workstation operator. Data is collected by human factors engineers who observe the users in the training scenario, administer questionnaires to elicit specific feedback, and conduct individual and group interviews following each phase of the evaluation.

Following the exercise, the human factors engineers, in coordination with the Training Effectiveness Advocate, extract from the feedback a set of user comments which are loaded into a database. A second analysis is then conducted to determine which of the user comments address technical issues relevant to CCTT.

The set of technical issues is then assessed by a review board comprised of CE team leads and chaired by the Chief Engineer. They jointly identify which issues are true deficiencies in the system implementation and need to be tracked to insure that they are remedied, which ones are clearly outside the scope of the program and belong to management for consideration as the basis for an engineering change proposal, and which ones are a planned future build functionality. Issues in the first category are reported to the CE team. They manage and integrate the appropriate hardware or software modification to address the problem and they document in a formal tracking system the results of their efforts.

RELATIONSHIP TO SYSTEM TESTING

The user evaluation approach taken in CCTT is designed to complement and support formal technical and acceptance test procedures. The user exercises themselves do not attempt to cover all test conditions because they focus on an operational scenario based on Army doctrine and training methods. However, since the exercises are designed based on the build capabilities they do evaluate a significant portion of the technical requirements met in a given build.

As the seven builds and associated user exercises progress over time, the tactical scenarios used are not redesigned, but rather they evolve through enhancements. This both saves scenario development time and assures that what is planned for can be accomplished because a similar scenario was used in the previous USEX. The scenarios which evolve will be recommended for use in technical testing such as PPQT and as the basis for operational testing acceptance.

Government test agencies have representatives assigned full time to the IDT in Orlando. These individuals work with the User Exercise Working Group on a regular basis. Test agencies also send personal to observe the user exercises and evaluate their agency's planned procedures for test data collection. Although, there is not yet approval for this concept, we believe that during the user exercises for the later builds, a cooperative testing and user evaluation effort will occur which shares scenarios and exercise results. Potential savings to the overall program costs could be significant. Working cooperatively in this manner extends the integrated development approach [6] one step beyond simply engineering organizations working together. Our concept encourages engineering organizations and evaluation agencies to cooperate by sharing processes, products and data.

EXPERIENCES TO DATE

As of August 1994 we have conducted the user exercises associated with the first two CCTT spiral system builds. Build 3 User Exercise is planned for November of 1994. Government support has been superb. The exercises have proven to be an appropriate way to acquire early feedback from users and a way to show progress toward completing the system the user community has requested.

Results of the User Exercise are reported in several forms. For USEX1 136 user comments were collected from which 75 technical issues were identified. For USEX2 86 comments, 27 of which were duplicates from USEX1. Table 1 summarizes the results of these first two user exercises and subsequent analysis. It is our contention that the 76 total Problem Trouble Reports (PTRs) identified as a result of the user exercises would have eventually been identified But this would probably not be until qualification testing when the cost to fix them would be considerably increased.

USEX	# User Cmts	# Tech Issues	PTRs
1	113	79	47
2	86	49	29

TABLE 1

All comments are captured in a database where their linkage to technical issues and final disposition is shown. This database has been incorporated into the CALS system used on CCTT so that all government and contractor personnel have access to results. A formal report is also prepared on each exercise, it includes a training effectiveness assessment. That assessment indicates, for each training tasks underlying the exercise scenario whether the task can be fully trained, partially trained, or not trained using the current build configuration.

We have found it necessary to refine the manner in which findings (user comments) are processed and reported. The initial concept of simply reporting these comments out to the engineering organization, specifically the CE teams, was not effective. The three analysis steps described in III above was the result of evolving this process so that it generated information in the form of specific technical issues that is understandable and could be acted upon by engineers.

It is important to manage expectations and that procurement agency concerns be continuously addressed. For the early builds, we are using Subject Matter Experts from the Army centers as our soldier subjects. In some cases, these are NCOs who have previously reviewed CCTT specifications and designs so they are intimately familiar with both what is being developed and the incremental build process. For Build 3 to 7 User Exercise we plan to use soldiers from field commands who are unfamiliar with the system as subjects. This will add a requirement for another type of expectation management. We have worked diligently to identify system anomalies that are the result of using early development (in some cases just prototypes) components which will be improved or replaced later in the program. User subjects receive a detailed briefing on these anomalies prior to beginning the training scenarios.

CONCLUSION

We plan to continue using the approach described above for all CCTT builds. Our first two user exercises have been well received and supported by the contractors involved, the government procurement agency (STRICOM), and the Army user community (TRADOC). The approach will be refined and institutionalized for future use on complex training systems.

Incorporating user evaluations into a development effort of any system is critical to the successful implementation of the human-computer interaction components of that system. This is a fundamental aspect of user centered system design and for software developed with a usability engineering focus. A training simulation system is primarily a human-computer interaction, all other components – modeling algorithms, terrain data bases, etc. – exist to support that purpose. For this reason, it becomes even more critical to conduct early and continuous user assessments of training simulations. For CCTT we have developed a methodology using tactically-based scenarios that successfully accomplishes that purpose.

REFERENCES

1. Johnson, W., Mastaglio, T. and Peterson, P., "The Close Combat Tactical Trainer Program", *Proceedings of the 1993 Winter Simulation Conference*, Los Angeles, CA, December 12-15, 1993, pp 1021-1029.

2. Thorpe, J. A. "The New Technology of Large Scale Simulator Networking: Implications for Mastering the Art of ", *Proceedings of 9th Interservice/Industry Training Systems Conference,* pp. 492-501, American Defense Preparedness Association, Washington, D.C., 1987.

3. Alluisi, E. A. "The Development of Technology for Collective Training: SIMNET, a Case History," *Human Factors,* 33(3), pp. 343-362, 1991.

4. Loral Federal Systems, *The Close Combat Tactical Trainer Operational Needs Document*, June 1, 1993.

5. Bimson, K., Marsden C., McKenzie, F., and Paez Naomi, "Knowledge-Based Tactical Decision Making in the CCTT SAF Prototype", *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation,* 293-301. Institute for Simulation and Training, Orlando, FL, 1994.

6. Mastaglio, T. W., and D. Thomson, "The CCTT Development Approach: Integrating Concurrent Engineering and User-Centered Development", *Proceedings of 15th Interservice/Industry Training Systems Conference*, pp 354-360, 1993.

7. U.S. Air Force, "Simulator Test 2000", Training Systems Program Office (SPO) at Aeronautical Systems Division (ASD/YW), Wright-Paterson Air Force Base, paper available from ASD/YW or first author.

DEFINING THE USER'S TRAINING TECHNOLOGY NEEDS: THE ARMY'S EXPERIENCE

Marta J. Bailey
Diana Tierney, Ph.D.
Headquarters, U.S. Army Training and Doctrine Command
Fort Monroe, VA

ABSTRACT

User acceptability of new technology is directly related to the degree to which the technology satisfies the user's needs. The salience of the relationship between user needs and user acceptability is underscored by the tenets of Total Quality Management (TQM). According to TQM philosophy, the technology user is defined as the *customer* and the appropriate role of the research and development (R&D) community is to satisfy customer needs. But, how knowledgeable is the training technology user of his own needs? Can trainers influence the course of technology development to maximize gains from their technology investment?

Conceptually, success in this endeavor requires the training technology user to have a strategic vision of where training is going in the next 5-10-20 years. The vision needs to be translated into technology requirements for the near-, mid-, and long-term. Finally, the requirements need to be communicated to the R&D community so work is focused on the identified goals.

The U.S. Army Training and Doctrine Command (TRADOC) has an effort underway to identify, prioritize, and communicate the Army training community's science and technology (S&T) requirements to the R&D community. In this paper, we discuss some of our experiences setting up this management process, interfacing with the R&D community and lessons learned. Clearly, the process requires communication between the users/customers and researchers to clarify requirements and identify useful directions for research. In addition, it is important to form alliances with users from other services, commands, and agencies. Lessons learned from our experiences so far indicate users need to be smart about what they need, be smart about science, work together, and be proactive in order to effectively manage technological change.

ABOUT THE AUTHORS

Ms. Marta Bailey is a Personnel Psychologist working for the Office of the Deputy Chief of Staff for Training at the U.S. Army Training and Doctrine Command (TRADOC), at Fort Monroe, VA. She manages the TRADOC Training Research and Development Action Plan which is designed to focus R&D efforts so the Army's future training strategies are supported by the R&D community. She has a Master's degree in Psychology from Old Dominion University and has worked for the Army for eight years.

Dr. Diana Tierney is a supervisory Personnel Psychologist and Chief of the Training Research and Studies Branch in the Office of the Deputy Chief of Staff for Training at HQ TRADOC. She oversees the Training Research and Development Action Plan and the Army training effectiveness analysis program. She has a doctorate in experimental psychology and completed a post-doctoral fellowship at the Institution for Social and Policy Studies at Yale University. She has worked for the Army since 1985.

DEFINING THE USER'S TRAINING TECHNOLOGY NEEDS: THE ARMY'S EXPERIENCE

Marta J. Bailey
Diana Tierney, Ph.D.
Headquarters, U.S. Army Training and Doctrine Command
Fort Monroe, VA

Imagine the world of the 21st century where a reserve component soldier in his Montana living room prepares for his first mission in a trouble spot half-way around the world. He is immersed in a virtual environment populated by computer-generated enemy forces operating under their current doctrine. Virtual crew-members interact with him as they traverse along terrain that accurately matches the anticipated location of the mission. He experiences the sights and sounds of the mission and feels the adrenaline racing though his body. He encounters difficulties but has an opportunity to react and test a variety of responses in the safety of his home. Evaluation protocols verify performance success so we are confident training has made him a seasoned soldier who is ready to face the challenges ahead.

CAN WE GET THERE FROM HERE?

Managing change is a phrase frequently heard among Army trainers today. The proliferation of new training technologies is one area where change is occurring at an astounding rate. One way the U.S. Army Training and Doctrine Command (TRADOC) is attempting to manage technological change in Army training is through the Training Research and Development Action Plan (TRADAP). This paper describes the rationale behind TRADAP and some lessons we have learned from initiating a management process to guide technology toward our goal---providing high quality training to soldiers in the near- and far-term

The mission of TRADOC is not only to train soldiers and leaders but to serve the function of "architect of the future." In its role as architect of the future, TRADOC writes the Army's warfighting doctrine and defines battlefield requirements for the future battlefield. TRADOC looks 5-10-20 years ahead to create as accurate a vision of the future state of the world as possible and define the doctrine, training, leadership, organization, materiel, and soldier requirements we will need for the Army to be prepared for contingencies in that future world. It is in this capacity that TRADOC has the primary role of interfacing with the research and development (R&D) community. TRADOC serves as the customer's representative by defining requirements and working closely with researchers to bring needed technologies to the ultimate customer, the soldier.

As training becomes increasingly reliant on high-tech methods, trainers see that they have an even greater stake in guiding the course of science and technology (S&T) investments to ensure that future training requirements are fully supported by R&D. Twenty years ago the Army training system relied primarily on training methods such as platform instruction in school houses, instructor demonstrations, practical exercises, training manuals, and field exercises. While these methods are still the mainstay of the current Army training system, we are increasingly transitioning to so-called high-tech training methods such as computer-based training, networked interactive simulators, and video-teletraining.

Many of the elements of the Army's future training strategies are embodied in the hypothetical opening scenario: training that is safe, realistic, accessible, cost-effective, environmentally sensitive, and versatile. These future training requirements present challenges at the same time advances in technology present tremendous opportunities to meet those challenges. The nature of training is likely to undergo a profound metamorphosis.

The potential for such changes raises a number of questions. How can Army trainers insure limited R&D dollars are spent on only the most essential identified needs? In what areas can we leverage training technologies from the private sector and on what areas should we target Army resources? How can we insure efficient transfer of new ideas and products to the user so important and costly discoveries are not "left on the shelf" where they will not benefit the soldier?

MANAGING TECHNOLOGY IS A MANAGEMENT ISSUE

Questions such as these are not unique to the Army or Army trainers. There are many academic and industry articles published that address these management issues. In this era of *re-engineering* and *reinventing government,* ideas on the management of technology in the market-oriented private sector may serve as a model to bring economies and efficiencies to federal government. In his 1991 article, Marc Hequet describes the relatively new field of "management of technology." The essence of the management of technology is bringing together managers, engineers, and scientists to reach a common understanding of their strategic vision, constraints, and technologies' potential. It should not be a revolutionary idea, but too often management and researchers have functioned independently of each other. Initiating interactions between managers and researchers to manage technology are mutually beneficial. An educated management can better plan for the future and when researchers understand that vision, the relevance of their work is enhanced.

TOTAL QUALITY MANAGEMENT OF TECHNOLOGY

To manage the proliferation of new technologies a number of writers suggest applying the Total Quality Management (TQM) philosophy. The language of TQM has permeated our culture in recent years, perhaps to the point of overuse. However, regardless of fad or fashion, themes of quality and customer focus are enduring. For example, Philip Francis (1992) effectively argues that the basic tenets of TQM can be applied to the management of R&D investments to make them more productive. At the heart of this perspective is the focus on the technology user as the customer. By understanding the customer's needs, the research investment can be focused on meeting those needs. Assumptions and guesses about customer needs must be replaced by direct knowledge based on close interaction.

This same focus on the customer is seen as the critical factor that translates into successful technology transfer. Frequently, the R&D community is separated from the users--the ultimate beneficiaries of their discoveries. Researchers tend to "throw their product over the wall and hope someone will catch it" (Wolff, 1989). Michael Wolff describes the key steps of successful technology transfer as beginning with user involvement up front. Rather than discovering you have a "solution looking for a problem," Wolff recommends active interactions between users and developers to explore actual problems, validate suspected needs, and educate users on what the new technology can do. From the early idea exchanges, user participation is needed at each step of the process (identify applications, package for user accessibility, train, and follow-up) to insure successful technology transfer.

TRAINING R&D ACTION PLAN

With the understanding that the Army training community needed to undertake a program to manage technological change systematically we adopted some of the prevailing private sector ideas in the development of TRADAP. Among these are creating a shared vision between the R&D and customer communities, engaging in active ongoing interactions, and following through to ensure successful technology transfer.

The purpose of the TRADAP is to ensure that efforts by the R&D community will enable TRADOC to build the essential technological foundation for mid-to-long range Army training requirements. The key activities associated with the plan so far are listed below.

1. Developed a prioritized list of 65 training technologies requiring research.

2. Met with Army R&D agencies to promote training research interests.

3. With aid of R&D agencies conducted technology assessment to determine status of identified research questions.

4. Initiated cooperative research endeavors with sister services.

5. Presented research requirements to industry.

6. Co-hosted two conferences on emerging training technologies.

7. Took steps to join with TRADOC combat development community to prioritize overall Army Science and Technology Objectives.

8. Explored organizational issues related to technology transfer and proposed organizational changes to facilitate effective technology transfer.

LESSONS LEARNED

Throughout the two year period of TRADAP development and execution we have learned a number of lessons that may be instructive to others who want to manage technological change and become smarter customers for the technologies that will shape their future. Thinking back over the chronology of TRADAP events certain accomplishments stand out as keenly important to the overall success of our S&T planning effort. What follows are recommendations for important steps to take and issues to consider in development of customer-based S&T planning efforts--recommendations derived from our experiences in trying to launch TRADAP successfully.

Lesson 1: Make S&T Planning Part of the Organization's Strategic Planning Process

On the surface it may seem obvious that organizations should consider S&T needs as they define future goals, missions and requirements. However, for the TRADOC training community this hasn't always taken place. Although some S&T needs have been identified by TRADOC, some have been identified by the R&D community, and some future S&T needs associated with future plans may not have been identified at all. Our TRADAP work has reinforced our belief that customers routinely need to consider major near-, mid- and long-term future organizational initiatives in terms of the underlying S&T requirements associated with each. We have found that even a very general consideration of S&T requirements provides an adequate starting point for discussions between the customer and the R&D community about the directions for future research. Our experience has also shown that in so far as the customer is able to present S&T needs clearly in the context of specific future organizational requirements the S&T needs are better understood and accepted by the R&D community.

Six of the key future directions for Army training, derived from TRADOC strategic planning documents, are listed in Table 1. There are numerous drivers for these changes including resource, environmental and safety constraints on large scale field exercises, the change from a threat to a contingency based Army mission, the high-technology battlefield, the move to consolidate some Army occupations, and the generally increasing complexity and difficulty of the jobs of soldiers and their leaders. Each of these factors has salient implications for the future of Army training and for the S&T advancements that will be needed to support training. Table 1 also presents a few example S&T areas TRADOC has targeted for development over the next 5-20 years. Note that each S&T area is directly related to one or more of the future training requirements.

Lesson 2: Make S&T Requirements Explicit

Identify S&T needs. One of the most difficult challenges facing the R&D customer is translating future mission requirements into enabling S&T developments needed to support those requirements. What does the Army's future need for realistic training that is environmentally sensitive, safe and accessible tell us about what, if anything, the research community must be doing today in the laboratory? In our experience, the surest way to answer that question is to open a dialogue between the customer's own experts in a given future requirement and the scientists

EXAMPLES OF TRAINING TECHNOLOGIES ASSOCIATED WITH FUTURE TRAINING REQUIREMENTS

Future Training Requirements	Key Training Technologies
• Provide accessible, cost-effective training that is environmentally sensitive, safe, versatile and realistic.	• Virtual reality • Knowledge of minimum essential simulator fidelity requirements that result in training transfer • Reconfigurable simulators
• Train leadership skills appropriate for any event over the range of military operations.	• Knowledge of complex decision making • Speech recognition technology • Methods to measure and enhance leadership performance
• Prepare leaders and soldiers to be adaptable and innovative.	• Artificially intelligent/expert system performance support aids • Training techniques to prevent/ameliorate negative effects of stress on individual and collective performance
• Train for contingency missions.	• Multiple scenario generation • Knowledge of how to best design, develop, and deliver "just-in-time" training • Collective performance measurement techniques
• Promote joint, combined, interagency perspective in training.	• Training and performance support aids for effective communications between joint, combined, or interagency forces • Joint service combat simulation integration • Determine organizational changes necessary to facilitate inter-organizational cooperation
• Modernize the training development and training delivery system.	• Development of training development expert systems • Knowledge regarding implementation and feasibility issues for various training media • Knowledge of effective learner preparation techniques

Table 1

the R&D community with the most expertise and interest in that topic.

It is essential that the customer bring at least general descriptions of future requirements to the table for discussions with the scientific community. If at all possible, the customer should also provide a list of best guesses as to research needed in the S&T areas supporting each requirement. This latter point is somewhat controversial in that some argue that defining research goals should be the sole province of the R&D community. However, our experience has been that thinking through the S&T associated with specific future goals not only makes us a better informed customer of R&D it helps keep us more involved with and smarter about the technologies that we ultimately must transfer to use within our own system. Further, we have found that the more specific we are in our thinking the more fruitful are our discussions with the scientific community. Cases in point have been TRADOC's two successful technology conferences co-sponsored with Army Research Office (on Virtual Reality Technology and Training, and Speech Recognition Technology and Training). At both Conferences trainers sat down with scientists to discuss future training and related S&T needs in these broad technology areas. The outcomes provided some clear guidance for future research in these areas (see Table 2).

Some of the Research Needed to Support Development of Virtual Reality Applications to Army Training
1. Visual display systems
2. Position sensing
3. Haptic interfaces
4. Software to create virtual worlds
5. Auditory displays
6. 3-D real-time interactive graphics
7. Behavioral representation
8. Human interface issues such as simulator sickness
9. Training transfer requirements
10. Speech recognition interfaces

Table 2

Establish S&T Priorities. Another facet of our TRADAP work has been establishing training S&T priorities from the customer perspective. This step can be taken once the customer has initially identified S&T research areas needed to support future requirements. The step is needed because it tells the R&D community what S&T the customer considers most important and where to focus scarce R&D resources. Table 3 presents a list of TRADOC's top 10 training S&T priorities based on rankings provided by key representatives of TRADOC's training and combat developments communities. We provided raters some criteria to consider in ranking research areas (e.g. likely payoff of research to Army training) and found that TRADOC staff were easily able to do the rank ordering. We were also pleased to find some good consistency between groups of raters (i.e. trainers and combat developers) on general areas of research considered most and least important. After all was said and done we felt relatively confident that many of the most important research issues had made it into the top of our list of priorities.

Narrow the Focus to Under-researched Technologies. Once S&T needs have been identified and priorities established a logical next step is to crosswalk S&T needs with research projects completed, on-going or planned by the R&D agencies. Our approach was to review research programs of key training R&D agencies and match up programs with our identified S&T needs. The creation of a database to sort R&D projects by our S&T requirements greatly facilitated our efforts. Once the crosswalk was completed we were pleased to find that the majority of S&T needs were met partially or wholly by on-going or recently completed research. In discussions with the R&D agencies about their ongoing projects the opportunities for technology transfer became evident and those areas in which little or no research had been done became the focus of our efforts to influence future R&D plans. An unexpected spin-off of this crosswalk was our ability as a customer to advocate for continued funding of research programs which our independent review had established as clearly meeting our needs.

TRADOC's Training R&D Priorities
1. Virtual Reality
2. Dynamic environment (terrain and atmosphere)
3. Embedded Training
4. Knowledge of fidelity requirements for training aids, devices, simulators and simulations
5. Combat development-training simulations
6. Simulation, integration, standardization
7. Reconfigurable simulator
8. Knowledge of skill decay for collective tasks
9. Effective technologies for training groups
10. Decision support technology

Table 3

Conduct Technology Assessment. Obviously, not every future initiative will require a foundation of new scientific knowledge or advanced technology. For example, one of TRADOC's near to mid-term plans is to explore cost-effective applications of distance learning technologies (e.g. video-teletraining) to the distribution of training to

Reserve Component units. Formative program evaluation and feasibility studies may be needed to prepare for this future change but much, if not all, of the actual R&D work on the required distance learning technologies has already been done. This example points out why technology assessment, determining the "state-of-the-art" for any given S&T area, is a crucial aspect of a customer's S&T planning process. If we can best meet a future requirement by using a mature "on-the-shelf" technology then the organization can focus energy on technology transfer and eliminate the often costly and time consuming R&D step. If the necessary S&T work has not been done then that must be where the initial emphasis is placed. We found that the R&D agencies and technologists within academia, industry and the Army's training community are a willing and helpful source of expertise to TRADOC for training technology assessment.

Lesson 3: Plan for Technology Transfer

Identify and involve customer sponsors. Of course the real payoff to the customer for good S&T planning is the availability of the new scientific information or technology advancements to upgrade operations--improve the quality of products, make processes more effective in meeting requirements, save resources by operating more efficiently, and avoid costs associated with outmoded products and practices. To reap this return-on-investment in S&T research the customer must actively participate in guiding and monitoring the S&T research from inception to completion.

Once the customer has identified and communicated top priority S&T research needs to the R&D community, the customer must identify the best customer representative(s) (i.e., research sponsor) to work each project with the researchers. Responsibilities of the sponsor will include at least: 1) working with the R&D agency to specify goals, objectives and expected outcomes for the research, 2) participation in periodic, regular reviews of research progress, 3) providing support and advocacy, if needed, for continued funding of the R&D project, and 4) initiating processes necessary for transfer of the technology to the prototype evaluation or feasibility study stage or direct integration into the system. Integration into the system may involve the sponsoring office in assisting with the rewrite of organizational guidance or policy governing operations (e.g. to integrate new scientific knowledge), development of training or job aids for users of the new technology, and obtaining funding needed to integrate new technologies across the system. Our experience suggests that the level of customer effort required to pinpoint S&T research needs is a small fraction of what customers must expend to transfer technology developments successfully. Yet it is easy for this crucial aspect of S&T planning to be neglected.

Gauge Organizational Commitment. We do not mean to suggest that sponsors can or should work alone to promote technology without the full involvement and support of their organization and its leadership. Rather, the orderly transfer of technologies needs to be an organizational imperative--a fully sanctioned and resourced aspect of the organization's mission and a recognized part of the organization's continuous quality improvement program. Our experience suggests that organizations, particularly those in resource constrained environments, are often so heavily involved in maintaining the current system that it can be difficult for them to put organizational resources behind future planning. Our recommendation is that before an R&D customer begins S&T planning they give full consideration to their organization's ability to plan adequately for and take the necessary steps to assimilate new S&T. If the commitment isn't there then the timing may not be right to assess S&T requirements.

Lesson 4: Form Partnerships With Other R&D Customers

One of the most fruitful strategies for us in development and execution of TRADAP has been aligning our efforts with those of other organizations with similar S&T planning goals. For example, TRADAP has been able to piggyback on the S&T planning and execution mechanisms developed by TRADOC's Battle Labs. The Battle Labs are actively involved in identifying TRADOC's S&T requirements associated with future battlefield operational capabilities requirements. Battle Labs have made great strides in developing processes for directly influencing the Army's S&T agenda and more generally communicating TRADOC's R&D interests. We have successfully joined TRADAP efforts with those of the Battle Labs in a number of areas including participation in the Battle Labs' yearly review of Army Science and Technology Objectives and participation in solicitations for industry science and technology developments. Another type of successful partnership for TRADAP has been joining with other organizations to pursue funding for S&T projects of mutual interest. For example, TRADOC is a participant in a Marine Corps led project to develop enabling simulation and virtual reality technologies for future training of military operations in an urban environment. We urge S&T customers to seek other customers to work with to develop effective strategies for interaction with the R&D community and to join with them to advocate for research in areas of mutual concern.

CONCLUSION

When we imagine that future world in which a soldier trains in a virtual environment we must keep one question in the forefront of our minds--How do we get from here to there? How do we achieve that envisioned end state, whatever it is? Our experience has led us to believe that a large part of the answer is active S&T planning by the organization responsible for creating that future state---the S&T customer. It is the customer who must set in motion and direct the series of events that will produce the required technological capability when its needed. This voyage to the future is far too important to leave the navigation to chance.

In this paper we have shared some of our perspectives on and lessons learned from experiences as a customer doing S&T planning. The experience has reinforced many of our beliefs about the value of organizations' attempting to manage technological change, the critical need for effective technology transfer, the importance of identifying S&T requirements associated with future plans, and the value of collaborating with other customer organizations in working S&T issues. We have also been impressed with how difficult it is, in terms of the time available, to pull in all the good S&T research ideas from the many knowledgeable and creative thinkers in our organization and how quickly future plans, future technologies, and hence S&T needs change. We have learned to our dismay that only a small fraction of the Army's R&D funding (about 2%) is devoted to research on the behavioral science issues of import to training, and that the White House has identified the lack of training and education R&D funding as a critical problem in this country. Perhaps most significantly, we have gained important new insights into how S&T advancements can potentially contribute to an exciting future for Army training. We recommend the S&T planning process to other customers as a means of gaining these insights and moving toward better management of technological change.

REFERENCES

Hequet, M., *Management of Technology,* Training, April 1991, Vol 28(4), p. 61-65.

Francis, P.H., *Putting Quality into the R&D Process*, Research-Technology Management, Jul/Aug 1992, Vol 35(4), p. 16-23.

Wolff, M.F., *Technology Transfer: a GM Manager's Strategy*, Research-Technology Management, Sep/Oct 1989, Vol 32(5), p. 9-10.

RESOURCE TRADE-OFFS FOR AVCATT
AVIATION COMBINED ARMS TACTICAL TRAINER

ALAN R. KELLER
DIRECTORATE OF TRAINING, DOCTRINE, AND SIMULATION
FORT RUCKER, ALABAMA

ABSTRACT

In times of acquisition budget constraint, we must show realistic trade-offs to justify future simulators. The Aviation Combined Arms Tactical Trainer (AVCATT) will permit critical collective training in exchange for minimal operating tempo (OPTEMPO) trade-off. Three approaches to determining resource trade-offs are presented: the Augmentation Approach, Futuristic Approach, and the Budget Constraint Approach. The break-even cost analysis for the Budget Constraint Approach reflects that AVCATT could pay for itself during its life cycle in exchange for an OPTEMPO trade-off of approximately one flying hour per crew per month. This trade-off translates to 26 operating hours of AVCATT per crew per month. Prior to deploying for "Operation Desert Storm," crews from the AH-64 Apache and OH-58 Kiowa equipped 2/229th, Attack Helicopter Battalion (ATKHB) trained gunnery and battle drill tasks in the Apache Combat Mission Simulator (CMS). Also during this time, companies trained combined arms collective tasks in eight reconfigurable simulators that were networked to form a collective training system. Using the collective training system, the company commanders were able to gain valuable unit cohesion before going into combat. In this scenario, the concept of simulation-based collective training passed the ultimate test--that of actual warfighting! The results of reconfigurable cockpit training experiments can be added to the Desert Storm evidence. Experiments involving 361 aviation officers reflect a need for a company level, combined arms collective training system, accessible to each battalion. During a time of defense budget constraints, the cost effectiveness of reconfigurable cockpits and reusable software must be considered in future acquisition strategies. Specifically, when the AVCATT acquisition effort comes before the scrutiny of milestone decision review officials, monetary savings and cost avoidances can be achieved by taking advantage of new simulation technologies. The time has come for not only accepting the cost and training quality benefits of simulation, but to also consider the AVCATT for both combined arms collective training and individual sustainment training.

Personal Biography: Mr. Keller is assigned to Headquarters, Directorate of Training, Doctrine, and Simulation, Fort Rucker, Alabama 36362. Phone: (205) 255-3096. A retired Army Major, he has held a variety of Special Forces and Airborne command and staff positions. His degrees are: B.S. in Management, University of Alabama; M.B.A. in Finance, Troy State University.

RESOURCE TRADE-OFFS FOR AVCATT AVIATION COMBINED ARMS TACTICAL TRAINER

ALAN R. KELLER
DIRECTORATE OF TRAINING, DOCTRINE, AND SIMULATION
FORT RUCKER, ALABAMA

INTRODUCTION

My research reflects that the AVCATT is the most cost- and training-effective alternative for conducting collective training in attack and scout helicopter units when used in combination with the actual aircraft. This combination will provide a fix to most training deficiencies presently existing in several Army aviation units. However, AVCATT must be purchased before we can use it for training, and herein rests the challenge. With acquisition budgets shrinking, how are we going to pay for AVCATT? Historically, commanders are hesitant to trade OPTEMPO for any training device because they consider the aircraft to be the best training medium. However, this paradigm may be changing as simulation technology becomes more advanced. What if simulation technology improves to the point that AVCATT, in addition to its collective training role, may also be able to train crew sustainment tasks as well as the Apache CMS? Three approaches to resource trade-offs are offered for consideration in the event trade-offs are the only way to obtain AVCATT.

TRADE-OFF APPROACHES

1. Augment Approach

The Augment Approach is actually an argument against trade-offs. It rejects the notion of OPTEMPO trade-offs and is based on AVCATT being an augmentation to the existing training program. It refers to the air combat aviator loss rate in WWII, which increased or decreased proportionately to the amount of collective training accomplished prior to the first air combat mission. Proponents of this approach believe that the trade-off for developing and buying AVCATT will be the saving of aviator lives during the first three to four missions flown in future air combat operations. Most commanders will agree that this is enough resource trade-off to justify the expense of AVCATT and, if we have to trade-off OPTEMPO, maybe we're better off without AVCATT! This theory's application to the modern electronic warfare battlefield is yet to be proven. The results of "Operation Desert Storm," concerning our readiness for initial attack helicopter operations without AVCATT, may be misleading. This statement is predicated on the credibility of the Iraqi threat at the time the battle was joined. After the numerous allied air sorties, many Iraqi units suffered heavy losses and ceased to function as an effective fighting force. Also, our forces conducted extensive pre-combat collective training in Saudi Arabia before joining the battle. The requirement for AVCATT is based on fixing training deficiencies that involve tasks which are not being performed adequately due to political, laser, ammunition, economic, environmental, or safety reasons. Augment Approach proponents state there is a minimum of flying to displace because certain collective tasks are not presently being performed during normal unit training periods. Therefore, they believe the primary role of AVCATT should be to augment, not displace, actual flying hours.

2. Futuristic Approach

2.1 Definition. Because the AVCATT is a crew, team, company, combined arms, and joint task trainer, it will help the commander conduct the unit training program prescribed by his Mission Essential Task List (METL). Proponents of the Futuristic Approach agree that AVCATT could displace a portion of actual flying a unit performs when performing collective training. They do not feel that the number of flying hours can be determined in advance, but must wait until after the AVCATT is fielded. After each unit has had time to become familiar with the device, an evaluation can be conducted to determine a more realistic amount of OPTEMPO to trade-off. They feel the individual unit commanders' preference for AVCATT will become apparent after fielding.

2.2 Determining Futuristic Trade-Offs. It will be difficult to draw Armywide conclusions because attack and scout helicopter units will have different collective training hours due to their unique missions. Thus, the number of flying hours saved will vary from unit to unit. My research was limited in that I could not visit all the different Armywide conditions under which attack and scout helicopter units are required to perform. However, I needed a general understanding of a typical ATKHB unit training program to determine trade-offs using the futuristic approach. I choose to study the 2/229th ATKHB at Fort Rucker for the obvious geographical advantages. My objective: to determine if an Attack Helicopter Company (ATKHC) commander, given certain training advantages, would be willing to trade-off actual flying hours for AVCATT hours. If he preferred AVCATT for a portion of collective training that he was presently conducting in the aircraft, this preference would surface these previously used flying hours for trade-off. In effect, I was trying to determine the best combination of AVCATT hours and actual flying hours from a unit commander's point of view. At least I would have a frame of reference of how this might be done in the 2/229th.

2.3 Approach. Consider a typical heavy company having an aircraft mix of five AH-64A crews and three OH-58 crews, all requiring Army Training and Evaluation Program (ARTEP)-Mission Training Plan (MTP) collective task training. In the 2/229th they would be authorized a maximum of 12 battalion training days and 36 company training days, for a total of 48 collective training days per year. It's probable they would fly three hours per training day. By multiplying 48 days times 3 hours per day we get a maximum of 144 hours annually for collective training per crew. If we multiply 144 hours times 5 Apaches we get 720 AH-64A flying hours per year. If we multiply 144 times 3

Kiowas we get 432 OH-58 flying hours per year. These flying hours were actually allocated in the unit's annual flying hour program and could have been used for collective training if the commander desired. Our company commander selected 21 ARTEP-MTP tasks that his company needed training in. Being without any organic or supporting Tactical Radar Threat Generators (TRTG), Army Development and Acquisition of Threat Simulators (ADATS), or Opposing Forces (OPFOR), he can only fully train 7 of the 21 tasks in his aircraft. The fact that an appropriate threat emission device is normally not available to aviation units on a weekly basis is a big disadvantage. AVCATT will solve this problem.

2.4 Findings. After coordinating with the Battalion S-3 on using all 48 training days, the commander decided to allocate equal training time to each task and fully train the remaining 14 tasks in AVCATT. He planned on flying for 16 training days, times 3 hours per day for a total of 48 hours. Multiplying 48 times 5 equals 240 AH-64A hours, and 48 times 3 equals 144 OH-58 hours. He determined he could realistically trade-off the delta, or 480 AH-64A flying hours and 288 OH-58 flying hours. From his perspective, he determined the best combination of AVCATT and actual flying hours to fully train all 21 of his selected ARTEP-MTP tasks. It is realistic to assume that all aviation company commanders would have different bottom lines because of their unique training requirements. In this approach, it is important to consider that AVCATT has already been produced and fielded. The question remains, "in times of budget constraint, how are we going to pay for AVCATT in the first place?" The answer to this question is discussed in the next section.

3. Budget Constraint Approach

3-1 Definition. The Budget Constraint Approach provides a method to determine resource trade-offs, in terms of flying hours, <u>before</u> AVCATT is purchased. For AVCATT the training developer's goal is to correct training deficiencies, increase unit proficiency, and provide a cost effective collective training system for our units. The total life cycle cost should represent the acquisition team's best shot at meeting this goal. This approach uses a break-even cost analysis, with the total life cycle cost as its break-even point, the device life expectancy as the schedule, and the using units flying hour OPTEMPO as the trade-off or billpayer.

3-2 Budget Constraint Approach Strategy. The strategy for trade-off of OPTEMPO begins with the zero-sum concept and flows through the acquisition stages to fielding the device. This is a total force strategy that pertains to reserve components as well as active units. As a unit gains AVCATT, its OPTEMPO is reduced by a predetermined amount. After AVCATT fielding, units will begin training on AVCATT and OPTEMPO savings will start to accrue. The break-even cost analysis reflects that AVCATT could pay for itself during its lifetime for a fairly small sacrifice in OPTEMPO. The OPTEMPO has been 14.5 flying hours per crew per month for all active Army aircraft since fiscal year 1992 and was requested in the defense budget for fiscal year 1995. For the same years, OPTEMPO for the Army Reserve and Army National Guard are 8.1 and 9.0, respectively. Assuming the OPTEMPO remain the same at the time of AVCATT fielding, they would only have to be reduced approximately 8 percent to 13.3, 7.5, and 8.3 to achieve the payback break-even point during the lifetime of the device. This reduction translates to 26 hours per crew per month availability of AVCATT.

3.3 Determining the OPTEMPO Trade-Off. The methodology for determining the OPTEMPO trade-off begins with the total life cycle cost for AVCATT less the Military Personnel-Army (MPA) cost since MPA is a sunk cost. The total life cycle cost (less MPA) represents the cost avoidance target that AVCATT must reach to break-even or pay for itself. The number of AVCATT company sets delivered per fiscal year was determined by the materiel developer. This information was used to determine the number of company sets in use per fiscal year, throughout the entire life cycle. Next, the number of active duty units, Army Reserve, and Army National Guard company size units that will be using AVCATT per fiscal year is determined based on a total force structure of 18 divisions, 4 corps, and 3 armored cavalry regiments. The number of AVCATT training hours, available per crew per month, can be calculated by multiplying the number of devices times AVCATT training hours per year to arrive at total hours per year. This figure is then divided by the number of units using AVCATT to arrive at hours per unit per year. This figure is then divided by 12 months to arrive at the number of hours per crew per month. Available hours per crew per month are calculated on the assumption that, when a company trains, all crews will train, and only one company will use a particular AVCATT device at a time. The figure of 26 hours per crew per month is the average availability for the total years used in the cost analysis. The annual OPTEMPO total cost, prior to units receiving AVCATT, is calculated by first summing the flying hour rates for the eight aircraft in each type company. The sum is multiplied by the flying hours per year (OPTEMPO rate times 12 months). This product is multiplied by the number of units using AVCATT to arrive at a total annual cost for each type unit. The AVCATT payback is determined by taking a percentage of these annual costs until the total reaches or surpasses the target. The cost analysis reflects that the break-even point is reached at approximately 8 percent of the total annual OPTEMPO costs.

COCKPIT FIDELITY COST/BENEFIT

1. Definition.

The subject of cockpit fidelity generally refers to the extent that simulator functions, equipment, and seating configuration resemble the actual aircraft. In this section we will have a general discussion on determining the most cost- and training-effective cockpit for combined arms collective training.

2. Cockpit Fidelity Approaches.

2.1 Three approaches are discussed: Reconfigurable Cockpits, Aircraft Specific Cockpits, and the Embedded Approach.

2.2 Reconfigurable Cockpit Approach. The first attempt to apply the concept of reconfigurability to aviation trainers was fathered by the Advanced Research Projects Agency (ARPA), formerly Defense Advanced Research Projects Agency (DARPA). This initiative was called Simulation Networking, or SIMNET. The original SIMNET program was designed for battalion level armored and mechanized forces and was the pioneer system for the Close Combat Tactical Trainer (CCTT) which was awarded in November 1992 to Loral, formerly IBM Federal Systems, as part of an estimated $120M contract. During experiments using the original SIMNET system, it was discovered that the tank icons would also fly. This led to the networking of aviation devices which were called SIMNET-AIRNET, or just AIRNET. In 1988, the first generation AIRNET cockpit was delivered to Fort Rucker and appropriately called the FRED--Fully Reconfigurable Experimental Device. It ran smack dab into an aviator mentality (mostly senior warrant officers) that soon became a paradigm that partially exists today-- "we must have a high fidelity cockpit system for AVCATT. Anything less will cause negative training transfer, and we will lose sustainment training credit." Thankfully, a few active and retired aviation commissioned officers disagreed. These visionaries realized that if the young Aviation Branch was going to play in the combined arms tactical arena, it needed an affordable networked system to perform combined arms collective training with its infantry and armor counterparts. They realized that an Army aviation high fidelity combined arms collective training system would cost nine times as much as the CCTT contract. The only chance for the AVCATT to become a reality rests with the opportunity to exploit industry Distributed Interactive Simulation (DIS) technology insertions, reusable software development, and reconfigurable cockpits.

* AIRNET. The original AIRNET seating configuration was shaped like a backward "L" and each cockpit allowed reconfiguration in minutes to accommodate a scout crew in the left and right seats, or an attack crew in the front and rear seats. The attack front seat could be very aircraft specific for gunnery training because it only has to accommodate the copilot/gunner. ARPA has turned the AIRNET devices over to the Army and the facility has been renamed the Aviation Test Bed (AVTB). Many improvements have been made since the arrival of the FRED in 1988. The AVTB is now the centerpiece for Army aviation warfighting simulation and is a Battlefield Distributed Simulation-Developmental (BDS-D) facility, on the Defense Simulation Internet. The Army's BDS-D program is pioneering the use of DIS technology. The AVTB has great potential as a facility but is woefully inadequate to represent modern aircraft on a realistic synthetic battlefield.

* Advanced Rotary Wing Aircraft (ARWA). The ARWA is a partially funded STRICOM program, started 18 months ago, to establish the AVTB as a world class research and development facility. The ARWA uses DIS technology, reconfigurable cockpits, and reusable software. It will allow for the replication of mission equipment packages of modern aircraft in a realistic synthetic environment with a high enough fidelity to satisfy BDS-D and training requirements. The ARWA will provide credible aviation force representation to support TRADOC Battle Lab Advanced Warfighting Experiments and warfighting simulations associated with the Louisiana Maneuvers. The resultant products of the ARWA aviation simulation research and development program may be inserted into the AVCATT and future aircrew sustainment trainers.

2.3 Aircraft Specific Seating Approach. The Aircraft Specific Seating Approach will cost in excess of $1M per cockpit. This cost becomes excessive because this approach requires a cockpit to be built for each type of aircraft used by the AVCATT target audience. If an ATKHC and an Air Cavalry Troop (ACT) are based at the same installation, each with different types of aircraft, a minimum of 13 specific cockpits will have to be purchased in order to train both units at different times and avoid building a second complex. A second complex should be avoided because it will require duplication of emulation stations, increase operations and maintenance costs, and require more Military Construction-Army (MCA) funds. If reconfigurable cockpits are used to train an ATKHC and an ACT at the same installation, only eight cockpits have to be built. For example, if the AVCATT fielding plan calls for 5 fixed buildings at Fort Bragg (x2), Fort Rucker (x2), and Wheeler Air Force Base, Hawaii, 48 aircraft specific cockpits are required. If reconfigurable cockpits are used, only 3 fixed buildings and 24 cockpits are required. Assuming both type cockpits each cost the same, the total cost avoidance when using reconfigurable cockpits over the aircraft specific seating cockpits is $101M. This cost avoidance includes $30M for the 24 additional cockpits and two additional buildings, and $71M in sustainment costs to staff, operate, and maintain the two additional fixed buildings with simulators.

2.4 Embedded Approach. The ultimate goal of Army aviation collective training developers is to have an embedded training capability in each aircraft, i.e., the actual aircraft cockpit is used for training. For example, in this approach the ATKHC will have a simulation software program to practice ARTEP-MTP tasks and gunnery team drills inside their actual aircraft. Aviators will be able to practice these collective skills when they are not flying. This embedded training approach is not intended to wear out flying parts, burn fuel, or turn rotor blades; however, the wear on avionics and other electronic systems must be considered. A mobile, auxiliary power source will be used instead of the aircraft engine. Crews may be able to use helmet mounted displays or be able to roll their aircraft into a dome where simulation can be used with the actual aircraft system and mission equipment package. Visual screens, using DIS technology, could then be positioned close to the cockpit. The crew's out-the-window view, from their actual aircraft cockpit, would then be a common simulated terrain data base. The aircraft's flight controls will have to be connected to field hardened computers so that the aircraft's functions and equipment will be appropriately responsive. This concept offers tremendous potential for peacetime and wartime collective training. The concept of embedded AVCATT collective training is easier discussed than accomplished. Much research needs to be done and Army aviation needs a world class facility to perform these experiments.

RECONFIGURABLE EXPERIMENTS

1. There is some evidence that aircraft specific seating is unnecessary for collective training effectiveness. A total of 361 captains and first lieutenants attending the Aviation Officer Advanced Course (AVOAC) classes 91-1, 91-2, 91-3, and 91-4 were sampled after they had performed 13 ATKHC ARTEP-MTP collective tasks for an average of 12 hours each in the AIRNET reconfigurable cockpits. This sample size was approximately 11 percent of Aviation Branch company grade officers at the time. Of the 175 attack or scout rated officers, 88% registered high marks for training effectiveness (See Figure 1). When asked to compare their own individual performance in the reconfigurable and the added cost if they were to use higher

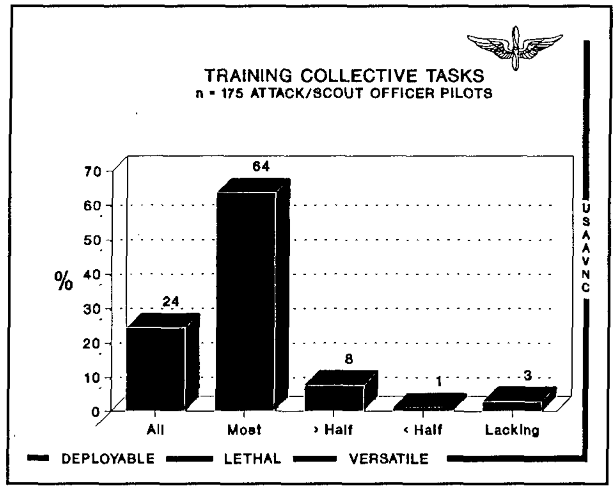


Figure 1

fidelity cockpits, 81% of the attack/scout officers sampled agreed that reconfigurable cockpits were cost- and training-effective and specific cockpits were not required (See Figure 2). The rationale seems to be that the added fidelity to attain aircraft specific cockpits is not worth the added expense; at least in this case, the reconfigurable cockpit was already effective for collective task training with some exceptions.

2. There seems to be some logical conclusions concerning the Reconfigurable Cockpit Approach:

2.1 Having a reconfigurable cockpit allows the commander to experiment with his mix of attack and scout mission aircraft to suit his training emphasis.

2.2 Air assault units located in the neighborhood of AVCATT sites would not benefit as much from specific cockpits as they would from reconfigurable. Realizing this is an attack/scout trainer, the AIRNET devices' proven utility for practicing air assault

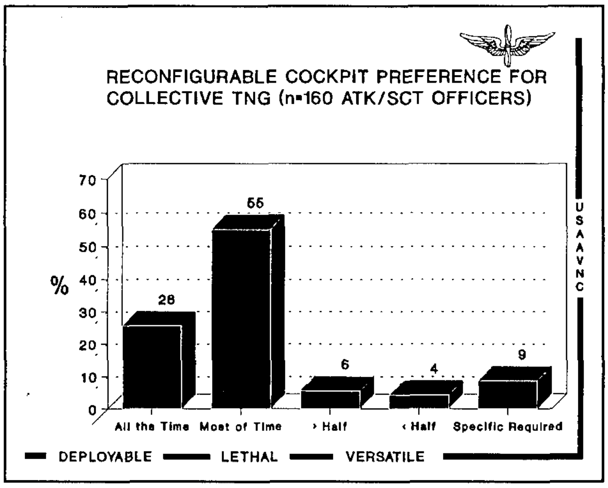


Figure 2

operations cannot be ignored.

2.3 The same attack battalion practical exercises now being performed in the AVTB could not be performed in high fidelity cockpits. Less than 20 percent of each class is attack rated.

2.4 There is considerable uncertainty with future funding for aircraft modernization and procurement. The AVCATT devices have a 20-year life cycle and will eventually require upgrades, however, the specific cockpits will become obsolete along with the aircraft. There may also be potential for specific cockpits to precede the aircraft. Should we build a Comanche specific cockpit for AVCATT before Comanche is approved for full-scale production? What if Comanche is further delayed due to budget constraint? Do we want to delay AVCATT indefinitely, waiting on a Comanche production decision?

2.5 To remain effective, the specific cockpit requires a technology update each time the aircraft is modified. Past experience indicates that the simulator modification will lag the aircraft modification. A similar, but not identical, simulator cockpit will maximize negative transfer. Therefore, in practice, a specific cockpit may offer more negative training as compared to a reconfigurable cockpit.

CONCLUSION

By the year 2000, the Army will construct and demonstrate a robust variety of synthetic environments to significantly improve simulation at all levels. Included will be the networking of manned virtual simulators like AVCATT, constructive models like Warfighters' Simulation 2000, and live simulation at the National Training Center. These efforts will fully network the corps level battlefield for simulation training.

During the time that we are trying to reach these goals, our resources are dwindling, and production of new systems is uncertain. We must consider the sacrifice of OPTEMPO to ensure that we have the best training environment during budget constraint and in the 21st Century.

The technology for simulation based training is arriving at breakneck speed. We must do our planning now to harness this technology fully.

LIST OF REFERENCES

1. Final Report, Aviation Combined Arms Tactical Training, Training Development Study-Phase II, USAAVNC, November 1991.

2. Defense Science Board (February 1976), Summary Report of the Task Force on Training Technology, Washington, DC, Office of the Director of Defense Research and Engineering, Department of Defense.

3. Defense Science Board (November 1982), Report of the Defense Science Board 1982 Summer Study Panel on Training and Training Technology, Washington, DC, Office of the Under Secretary of Defense for Research and Engineering, Department of Defense.

4. Defense Science Board (May 1988), Report of the Defense Science Board Task Force on Computer Applications to Training and Wargaming, Washington, DC, Office of the Under Secretary of Defense for Acquisition, Department of Defense.

5. Simulation and Training Research Symposium (April 1989), The Promise of Interactive Networking: New Levels of Training and Readiness in Peacetime, Orlando, FL, Dr. Earl A. Alluisi, Assistant for Training and Personnel Systems Technology, Office of the Under Secretary of Defense for Acquisition.

6. Training Developers Procedural Guide, Volume IV, Nonsystem Training Device Study Process, ATSC, April 1990.

7. Field Manual 25-100, Training the Force, November 1988.

8. Field Manual 101-5, Staff Organization and Operations, May 1984.

9. OH-58D Aircrew Cost and Training Effectiveness Analysis (CTEA), TRADOC, September 1986.

10. TRADOC Pamphlet 25-33, Army Training Glossary, 28 April 1989.

11. TRADOC Regulation 350-32, The TRADOC Training Effectiveness Analysis (TEA) System, 26 March 1990.

12. TRADOC Regulation 71-3, TRADOC Evaluation, Test, and Experimentation, August 1989.

13. Army Regulation 70-1, Systems Acquisition Policy and Procedures, 10 October 1988.

14. Capability Assessment of AIRNET-D for Combat Developments, USAAVNC, April 1990.

15. Scout/Attack Team Test Study: Technical Report, McDonnell Douglas Helicopter Company, April 1988.

16. Aircraft Survivability Equipment/Identification Friend or Foe, Institutional Training, Preliminary Training Development Study, USAAVNC, March 1988.

17. Test Report, Close Combat Tactical Trainer, Force Development Test and Experimentation, TEXCOM, August 1990.

SOURCE DATA ACQUISITION FOR THE CLOSE COMBAT TACTICAL TRAINER (CCTT) INTERSERVICE/INDUSTRY TRAINING SYSTEMS AND EDUCATION CONFERENCE

Dr. Robert H. Wright, Resource Consultants, Inc.
3051 Technology Parkway, Suite 280
Orlando, FL 32826-3299

BIOGRAPHICAL SKETCH

Dr. Wright is a Project Director with Resource Consultants, Inc. For the last four years he has been directly involved in identifying and analyzing DoD equipment data bases and for the last three years he has been responsible for the Close Combat Tactical Trainer Data Collection program. Dr. Wright can be reached at RCI 3051 Technology Parkway Orlando, Fl 32826 (407)282-1451.

ABSTRACT

With the DoD reduction in funds for the research and development of major weapon systems and the need to continue training soldiers under austere funding constraints, the need for simulators like the Close Combat Tactical Trainer (CCTT) becomes even greater. Information is vital to the effective and economical development of training aids, devices and simulators. As a part of the Army's information management initiative, the Simulation, Training and Instrumentation Command (STRICOM) through its support contractor, Resource Consultants Inc. (RCI) has taken the lead in collecting, collating, recording, storing and distributing information vital to the production of the CCTT. This data will be re-used in the development and procurement of follow-on trainers.

To build training devices like CCTT, the production contractor and the various Government agencies responsible for verification, validation and accreditation of the devices must have detailed data concerning the weapon systems that are to be modeled. To support this data collection requirement, RCI has developed four user oriented data bases. This paper discusses these data bases: the Document Cataloging System (DOCATS), the Equipment Characteristics Data Base (ECDB) the Combined Arms Tactical Trainer Task (CATTASK) data base, and the CATT Tracker data base. The tremendous cost and schedule savings that accrue by having data available at contract award make this approach viable for follow-on Combined Arms Tactical Trainers as well as other simulations or simulators that need data.

SOURCE DATA ACQUISITION FOR THE CLOSE COMBAT TACTICAL TRAINER (CCTT) INTERSERVICE/INDUSTRY TRAINING SYSTEMS AND EDUCATION CONFERENCE

Dr. Robert H. Wright, Resource Consultants, Inc.
3051 Technology Parkway, Suite 280
Orlando, FL 32826-3299

OVERVIEW

Information is vital to the effective and economical development of training aids, devices and simulations. As a part of the Army's information management initiative, the Simulation, Training and Instrumentation Command (STRICOM) has taken the lead in collecting, collating, recording, storing and distributing information vital to the production of the CCTT. This data will be reused in the development and production of the follow-on Combined Arms Tactical Trainers family. These training devices include the Fire Support Combined Arms Tactical Trainer (FSCATT), the Aviation Combined Arms Tactical Trainer (AVCATT), the Engineer Combined Arms Tactical Trainer (ENCATT), and the Air Defense Combined Arms Tactical Trainer (ADCATT). Other simulations such as WARSIM 2000, and training devices such as the Advanced Gunnery Training Systems (AGTS) and the M1 Maintenance Trainers will also have this data available. To support this massive amount of data, the Project Manager for Combined Arms Tactical Trainers (PM CATT) has directed development of several data bases to provide the data needed by the production contractor and other Government agencies during the development, production, fielding, and lifecycle support of the CCTT.

The CCTT is a group of fully interactive networked simulators and command, control and communications work stations. CCTT replicates the vehicle and weapons systems of a mechanized infantry or armor company team. CCTT portrays supporting combat, combat support, and combat service support elements and operates on a simulated real-time electronic battlefield. CCTT is a force-on-force free play simulation that provides for comprehensive after action reviews. The simulation will be at fixed locations and in a mobile configuration to train both active component, Reserve and National Guard units. The CCTT is a follow-on to the SIMNET-T with more battlefield effects, greater fields of view, open systems architecture and higher resolution terrain data bases.

CCTT is the first of a family of simulations that will eventually work together having a common communications protocol and data bases for the information needed to model the weapon systems in the field.

BACKGROUND

To build training devices the production contractor, and various Government agencies responsible for quality review of the devices, must have detailed data concerning the weapon systems that are to be modeled. Types of data include such things as vehicle dimensions, performance parameters (i.e., braking distances for a given soil type and speed), stochastic and deterministic failures, doctrine and tactics, etc. This data comes from sources such as specifications, test reports, engineering drawings, technical manuals, field manuals, video tapes, software programs, etc. This vast amount of data is currently stored in various media at multiple locations. Historically, the Government has relied on the production contractor to obtain whatever data the contractor thought was needed from whatever source the contractor could use. This resulted in obtaining data from various sources but there was no method to unify or validate the data or the sources. Using this historical approach also resulted in the Government purchasing the same data for weapon systems each time a new trainer was to be built. This has been very costly and resulted in little if any data management in training simulations developments. STRICOM has established a library that brings this information from diverse organizations and sources to one central location. The PM CATT decided that it would be cost effective and low risk to provide this data to the contractor as Government Furnished Information quickly. The contractor is still responsible for the collection and use of the data according to the contract, however, the Government has provided a single source for the data.

Providing accurate, timely and Government approved data to the production contractor has been shown to reduce costs and improve the schedule while providing data from a central source that can be used across programs. As a part of this effort, STRICOM has established a process to insure that the data provided to the contractor is reliable, timely, accurate and verified by Government sources. A data assistance office is responsible for collecting, processing and providing the data needed by the contractor. The data collection process started for CCTT almost one year prior to contract award. This allowed identification of types of data and primary and alternate sources for data. This permitted collection of a great deal of data before the contractor was even selected.

The first task was to identify the types of data required by the contractor to build CCTT. This was accomplished based on a thorough review of the CCTT specification by subject matter experts that had experience in the Army and with writing specifications. A detailed matrix was developed that listed the major categories of data for: weapon systems, terrain, weather, organizational structure, tactics, etc. This list included performance terms for vehicle dynamics such as engine and transmission rates and noises under various terrain and weather effects for example. The major categories included: organization, performance terms for control responses, performance terms for dynamics, performance terms for physical and dimensional characteristics, performance terms for sound, damage and failure data, repairs and maintenance, command and control and visual effects.

Once the major categories were determined research was undertaken to identify applicable data sources for each category and weapon system. This involved telephonic requests, written requests and visits to over 30 different Government and contractor agencies. While this research was underway, requests to the Defense Technical Information Center for data searches for each of the weapon systems in CCTT were made. Based on the results of these searches, selected test reports for specific vehicles were requested to determine if these reports had the necessary data. Also, requests for technical manuals and field manuals through the Army's Adjutant General Publications system were made.

Prior to contract award, the PM CATT held a data providers meeting to clarify which agencies would provide data and to obtain commitments from each agency as to when the data would be provided. After contract award detailed requests for data from the CCTT contractor were received. Those requests that couldn't be handled with the data already available were then called in to the applicable data providers. Since contract award there have been two other data providers meetings to further define the data to be provided.

Sources of this data include various Army Materiel Command agencies such as the Project Managers for the weapon systems, the Army Materiel Systems Analysis Activity, the Service schools, the Training and Doctrine Command's Analysis Activities at Ft. Leavenworth and White Sands, and the Test and Evaluation Command to name a few. Data that is provided to the library is received, abstracted and cataloged in a data base called the Document Catalog System (DOCATS). Abstracts from the data are sent to the responsible agency for review to insure that the data is current and accurate. Periodically the data certification/approval committee, chaired by the TRADOC System Manager, reviews each of the abstracts to verify that the data is correct and current.

DATA BASES

To provide this data to the contractor and other Government agencies in a timely and user friendly manner required development of four different data bases with unique capabilities. Each of these data bases were designed to meet specific needs of the CCTT program. Their structures are applicable to not only the follow-on CATT programs but to other Army and other Service programs that need data for their models and simulations. All four data bases are written in FoxPro 2.5. The data bases operate on a Local Area Network (LAN) and can be reached via either a dial-in modem connection or via the STRICOM LAN. Users on the LAN can go from one data base to another via keyword links. These data bases are closed systems: that is, data can only be accessed by authorized users. This prevents unauthorized modifications to the data and provides for traceability from the source data to the final product. Access to the data bases via the LAN is controlled through the PM CATT. Copies of the data bases can be obtained through the Tactical Warfare Simulation and Technology Information Analysis Center (TWSTIAC) at the University of Central Florida in Orlando, Florida.

DOCUMENT CATALOGING SYSTEM DATA BASE

The first data base is the Document Cataloging System or DOCATS. This is a data base of all the documents in the library. The library has over three thousand documents but is not limited to only paper copies. The library contains aperture cards for various weapon systems such as the M1 family of tanks; video tapes of various weapons and tactics, sound recordings of various weapon systems, and computer programs for simulations such as CASTFOREM. The library also serves as a pointer to other DoD libraries such as the Defense Mapping Agency's data bases. Titles and abstracts of the source documents are maintained and updated using FoxPro 2.5. As each document comes in to the library, the document is abstracted, cataloged, entered into the DOCATS and then shelved. User requests for documents are processed daily. When a request is received, the document is pulled from the shelf and sent out for reproduction. Within 24 hours of the request, the document has been provided to the user.

The DOCATS provides the document name, author name, date, abstract, key words list, and other pertinent information about the document. This information includes: classification level, approval status, source(s), etc. The program also lists the unique document number and date of publication. Users access the DOCATS via a program written in FolioViews, a hypertext editor. The DOCATS application is available in a runtime version on one diskette, or via the LAN. Users can browse the list of documents, search using key words or search by weapon system name. Boolean operators can also be used to narrow or broaden the search. Whatever the search technique, the program searches the title, key words, abstract and notes fields to find matches. All hits are then made available for review by the user. The user can also print out the list of documents that are pertinent to the search. Links are available from DOCATS to the other data bases.

CATTASK DATABASE

The next data base is the CATTASK data base. This data base was designed to provide training data from task manuals, soldier manuals, How To Fight manuals, subject matter experts and training studies into one central source. The intended audience for this program is the software engineers that need to know how a specific unit conducts its missions, or how individual soldiers do their soldier tasks as a part of the weapon system or the unit. The CATTASK data base is based on the Mission Training Plan (MTP) manuals that outline collective tasks performed by specific units. These collective tasks are represented by various battlefield operating systems such as maneuver, mobility, air defense, command and control, etc. The collective tasks are directly linked to the task standards, situation training exercise, tactics, techniques and procedures, missions, training mode evaluation and individual tasks making up the collective tasks and subject matter descriptions.

One of the primary uses for this data base is to provide a clear traceability tool to show that the software code for the semi automated forces is based on current and correct doctrine. This data base was written assuming that most software engineers writing this code had little or no military experience and needed detailed descriptions of exactly what actions took place for all fire and maneuver elements. To help in this effort the CATTASK data base also provides diagrams from approved field manuals that describe the various actions that units must be able to conduct on the battlefield. In the near future video clips will be added to the data base so that the programmer can see and hear a unit conducting formations. The screens for the CATTASK data base are menu driven and allow the user to conduct searches very easily. Searches can be conducted by organization, by MTP task, by battlefield operating systems, by mission or by situational training exercise. Once a task has been identified, the program lists the applicable source, the task, subtasks and standards associated with the task. The user can then choose to review the table of organization and equipment associated with the task or choose the appropriate reference that describes how that task is to be done in terms of conditions and standards. A complete list from the field manual or MTP describes in detail how the task is to be done. In addition, graphical depictions of the task from the field manual or other approved source can be displayed.

The CATTASK program also lists the individual tasks associated with each collective task and subtask so that a complete source of data is provided in one program. In addition to this information, the CATTASK program incorporates the Combat Instruction Sets that have been developed by the production contractor. These are in-depth, detailed descriptions written by subject matter experts that identify the "how to" details not mentioned in the field manuals and other sources. This program also links similar tasks from various MTPs so that comparisons and analyses can be accomplished. The production contractor will be using this program to provide the information needed for the programmers and to provide traceability, since blocks of code written for a task will be tagged for each task. This will allow using the same block of code for similar tasks. For example, to conduct a hasty attack for a mechanized infantry platoon involves many of the same tasks and subtasks that a tank platoon must perform. Once the code is written for the semi automated forces to do the task for the infantry platoon, the same code could be reused for the tank platoon. Currently 40 MTPs related to mechanized infantry and tank company from operations are in the CATTASK data base. Like DOCATS this data base is accessible through the STRICOM LAN, via modem or on disk through the Tactical Warfare Simulation and Technology Information Analysis Center.

EQUIPMENT CHARACTERISTICS DATA BASE

The third data base is the Equipment Characteristics Data Base (ECDB). This data base, stores and retrieves physical and performance characteristics for weapon systems that will be used by engineers and programmers to develop realistic manned modules and semi automated forces. The information is organized into a collection of weapon systems and their physical and performance characteristics. For example, engineering drawings, audio tapes and performance data such as speed over various types of terrain will be in this data base. CAGE Code and part number detail for the CCTT manned modules will also be available. Each performance characteristic may be decomposed into related sub-subclasses or other performance characteristics. Parameters and their related values are at the lowest characteristic level. Each individual value may be dependant on conditions for its validity. All this information is presented to the user in a user friendly windows type interface. Data to populate this data base comes from approved test reports, technical manuals and technical data packages provided by the Project Managers for their respective weapon systems. The exact data requirements for each vehicle are still being decided. This data base is still under development but it is populated with some data for the M1 and M2 vehicles and is operational on the LAN. The ECDB can also link to the other data bases via the LAN by weapon system name or TO&E.

The opening screen presents various choices that the user can select simply by clicking on the appropriate area. Detailed descriptions of the weapons systems are provided as well as views of the vehicle. This data is helpful in identifying parts and major sub assemblies of the weapon system as well as identifying the drawing numbers for various components. There are also Initial Graphics Exchange Specification (IGES) drawings that will be available. Other information includes such things as speed, fording depth, turning radius, braking distance curves, etc. This data base is still being modified and won't be fully operational until summer of 1995.

CATT-TRACKER DATA BASE

The fourth data base is an extension of the DOCATS data base designed to identify technical information by system, platform and functional user category. This is the CATT-TRACKER data base. The technical document requirements are identified and cataloged for each of the functional categories. The document identification number, status, location, effective date and DOCATS library call number are presented to the user on-line or in a report format. This program is designed to be a management tool used by the Project Manager to track the status of the data collection effort. The program resides on the LAN and can be reached via modem. Each of the functional categories provides the user with the required technical data for developing simulated systems for manned modules, work stations or semi automated forces at the weapon system level.

BENEFITS

The advantages to having this kind of data available online include reducing the time and money spent to collect the data separately for every new program. These data bases provide timely, accurate and approved data that contractors and other Government agencies can use today. Companies interested in access to these data bases must contact TWSTIAC or the PM CATT directly for permission. The benefits to contractors include the ability to go to a single source to get approved data. Another advantage is that this system is available via modem or on diskettes. In fact, the DOCATS library system has been made available to potential bidders interested in the M1 Maintenance Trainer, Advanced Gunnery Training System and WARSIM 2000 as a source for publications the Government intends to make available for these programs. Soon, the CATTASK data base will be available on CD ROM making access easier for potential users.

CONCLUSION

In conclusion, the need for accurate, correct and timely data to support the Close Combat Tactical Trainer program and other comparable programs coupled with the advances in technology make development and use of data bases like DOCATS, ECDB and CATTASK essential to the production process. The open architecture design of these systems coupled with their building block design make it easy to modify these data bases for any particular need or specific program. Also, the tremendous cost and schedule savings that accrue to having data available at contract award make this approach viable for not only follow-on CATT programs but for other programs that need comparable data. The STRICOM initiative to provide this data will continue to reap benefits for many years to come not only in reducing costs, but in providing a means to trace computer code back to the original source, thus providing for a more reliable validation and verification process and ultimately providing the soldier in the field with the best product available.

THE CHALLENGE OF MANAGING DOMAIN ENGINEERING

Glenn W. Dillard
Naval Air Warfare Center Training Systems Division
Orlando, FL

Michael R. Welch
RDR, Inc.
Orlando, FL

ABSTRACT

The shift of DoD software development practices toward the megaprogramming paradigm is creating new challenges for project management. Megaprogramming is a twin lifecycle paradigm which separates Domain Engineering from product acquisition. While the product acquisition lifecycle is not new territory for project managers, the Domain Engineering lifecycle requires fundamental changes in technical management and organizational practices. Some of the challenges raised by Domain Engineering are: adopting a product-line focus, planning Domain Engineering, establishing a Domain Engineering organization, staffing a Domain Engineering organization, and integrating Domain Engineering from several suppliers.

Unlike the classical software development lifecycle, the products of the Domain Engineering lifecycle persist beyond the lifecycle of any single product. This effect creates the opportunity for leveraged reuse between products; the purpose of megaprogramming. However, the persistent nature of Domain Engineering products has naturally motivated the customer to take a much more active interest in their formulation, including even active participation in the Domain Engineering process. While understandable and even desirable, this interest has raised additional Domain Management challenges in re-balancing traditional customer/contractor relationships and managing joint organizations.

ABOUT THE AUTHORS

Mr. Glenn Dillard is a lead project engineer with the Naval Air Warfare Center Training Systems Division. He has a wide variety of experience in training and weapon systems development programs, both in industry and civil service. Since 1983, he has worked on naval training simulator projects including submarine, aviation, electronic warfare, and intelligence simulation projects for NAWCTSD. He is currently the Lead Project Engineer for the Navy/STARS Demonstration Project in the Air Vehicle Training Systems domain. Mr. Dillard holds a Bachelor of Science degree in Electrical Engineering from the University of Central Florida.

Mr. Michael Welch is a Senior Software Designer with RDR, Inc. He has developed submarine and aviation simulated trainer systems for the past 10 years. He has been involved in software design and development for 27 years. Mr. Welch holds a Bachelor of Arts degree from Michigan State University and a Master of Fine Arts degree from Tulane University.

The Challenge of Managing Domain Engineering

Glenn W. Dillard
Naval Air Warfare Center Training Systems Division
Orlando, FL

Michael R. Welch
RDR, Inc.
Orlando, FL

INTRODUCTION

The Challenge

Domain Engineering is a new approach to product development. When applied to the development of software, Domain Engineering offers the promise of improved cost and quality control through the inherent reusability of planned product-line software assets.

Because Domain Engineering follows a new and different development path, management of a development process using this technique must be open to the adoption of new strategies and practices. Use of this technology presents challenges and opportunities; effective management of the process is the greatest of the challenges.

This paper will contrast the classical approach versus the Domain Engineering approach to software development management. An analysis of unique Domain Engineering management challenges will be provided. Recommendations for meeting challenges based upon the Navy/STARS Demonstration Project experience will also be presented.

Background

The DoD currently mandates that software developed for use on DoD projects follow the classical development cycle defined by DoD-STD-2167A; Domain Engineering adheres to the process documented by the Software Productivity Consortium's (SPC) Reuse-driven Software Processes Guidebook and spirals down an evolutionary path to achieve its goals.

Classical Waterfall Model

The DoD prescribed model for software development is described as a waterfall because

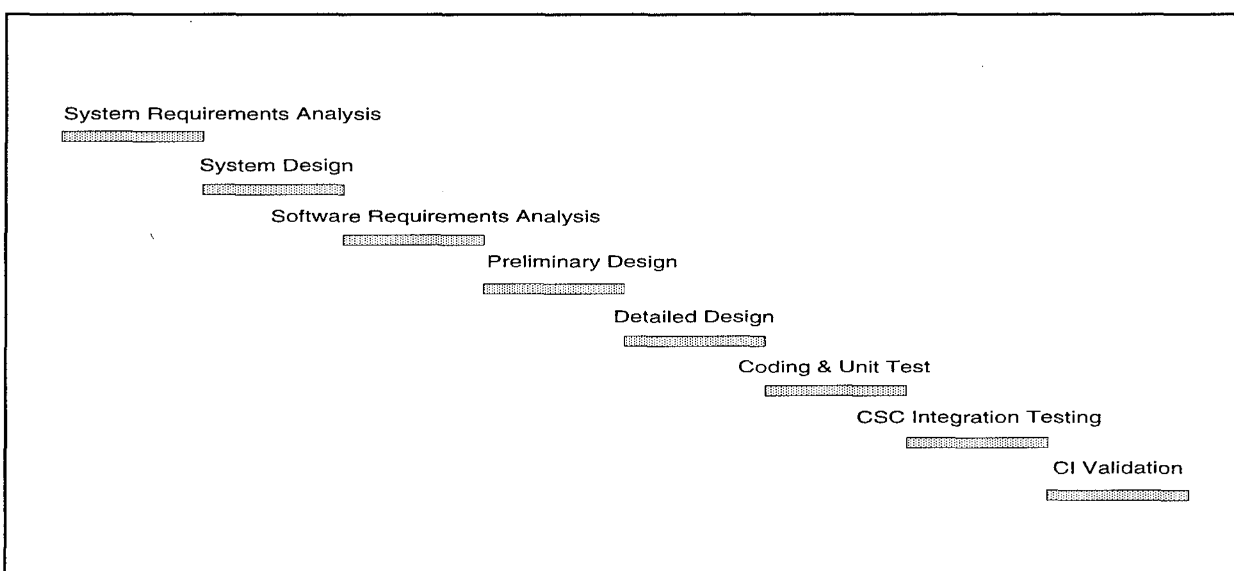


Figure 1 - Waterfall Process

the output from one step becomes the input for the succeeding step. Figure 1 illustrates this notation. The work effort flows from the top downward to the right until the project is complete. As part of this discrete step process, a formal review is placed as a gateway between process steps. A succeeding step may not be started until the gateway review is held and all objectives are met.

This linear development model precludes the effective use of partial or intermediate solutions to design problems. Any model which might provide insight during design is a throw-away investment. Development of the actual product is undertaken in the appropriate steps, at the appropriate time and by the appropriate technical group.

Scope

The Navy/STARS Demonstration Project is a research and development project undertaken as part of an Advanced Research Project Agency (ARPA) sponsored software program chartered to measure the benefits of a software development concept termed megaprogramming. Megaprogramming is defined as a domain-specific, process-driven, reuse-based, technology supported way of developing software intensive systems. The Navy has chosen modeling and simulation of Air Vehicle Training Systems (AVTS) as its domain.

The Navy/STARS Demonstration Project is based upon the two-life cycle process of Domain Engineering and Application Engineering versus the traditional single-life cycle product acquisition. The baseline for the process is derived from the concept of Synthesis promulgated by the Virginia Center of Excellence/SPC, illustrated by Figure 2.

Application Engineering is a standardized process by which projects produce and deliver specific system applications to a customer. In terms of objectives, it is equivalent to conventional "one of a kind" software development. Application Engineering focuses on requirements and engineering decisions that are sufficient to describe a particular system, given a family of systems. Deliverable work products, including code and documentation, are derived from these decisions using adaptable forms of the work products developed by Domain Engineering.

Domain Engineering develops the standard Application Engineering processes which support the decision making and the work product creation required by an application engineer. These processes, derived by Domain Engineering, support a product-line business

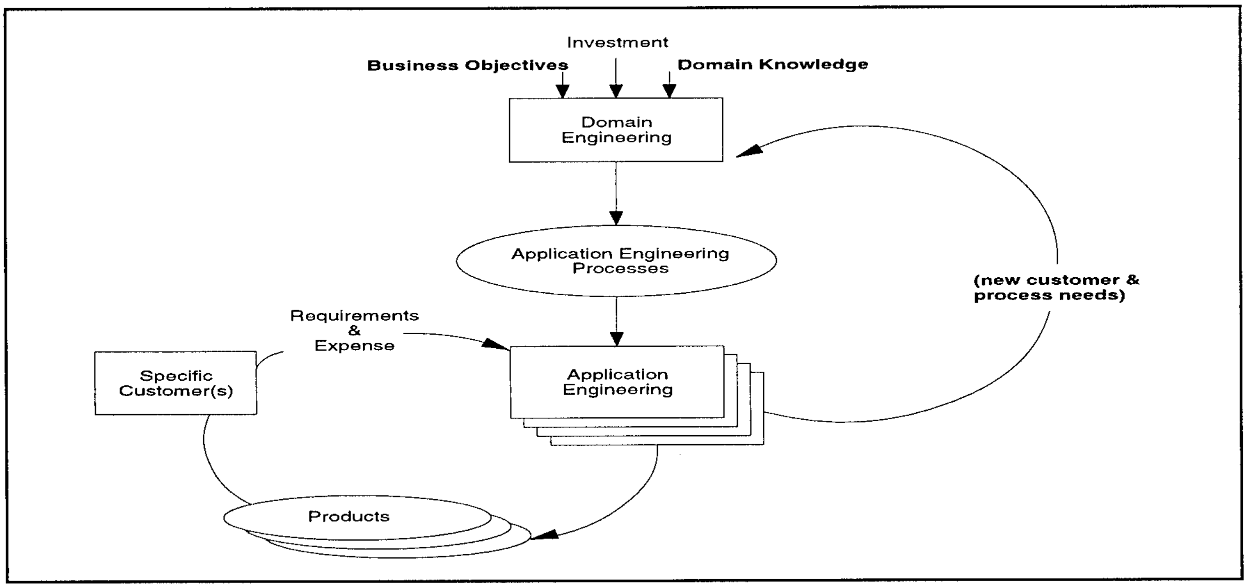


Figure 2 - Two-Life Cycle Paradigm

case of multiple applications.

Throughout the Navy/STARS Demonstration Project, the challenges of Domain Engineering have been and will continue to be addressed, implemented and evaluated with the goal of iteratively improving all Domain and Application Engineering processes.

Evolutionary Spiral Process Model

The strategy underlying the Domain Engineering process is a spiral process model that encourages the evolutionary solution of product-line problems and the development of product-line designs. The Domain Engineering process is followed iteratively touching upon each of the activities defined by Figure 3. A single evolutionary spiral includes an understanding of the domain context, an analysis of risk avoidance, a plan for iteration development, development of the product and a final evaluation of processes, products and plans. At this point, the iterative spiral can begin again with the addition of changes to the preceding spiral.

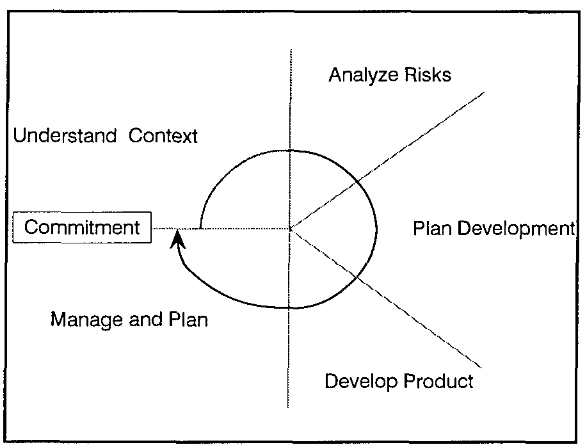


Figure 3 - Evolutionary Spiral Process

A primary benefit gained from the iterative spiral is the ability to utilize solution prototypes to evaluate new approaches to problem solution and design derivation. Each successive cycle builds on the process and product frameworks developed during the preceding iterations. By expanding and evolving processes and products, no intermediate work is lost. Additionally, design reviews may accompany working demonstrations of the partial products during the commitment phase of the spiral.

CLASSICAL SOFTWARE ENGINEERING MANAGEMENT

The emphasis for classic software engineering is set at the very beginning of a project with the customer's requirements specification. The overall engineering and management goal is to satisfy the unique requirements posed by the specification. In most cases, the specification is not reuse oriented; rather it is the documentation for a custom-built product at a custom price!

Software Development Process

The classical waterfall approach to software engineering and software management is driven by the products created as part of each step. As Figure 1 indicates, eight steps of product creation and their associated reviews carry the original concept from system analysis to product certification. These eight discrete steps are further subdivided by general engineering disciplines:

- System Engineering -
  1. System Requirements Analysis
  2. System Design
- Software Design -
  3. Software Requirements Analysis
  4. Preliminary Design
- Software Development -
  5. Detailed Design
  6. Coding and Unit Testing
- Software Testing -
  7. Component Integration Testing
- System Testing -
  8. Configuration Item Validation

With the division of responsibility parceled out to separate functional disciplines, each group performs its tasks uniquely. Software management is left to tie together process, product, cost and schedule.

The formal reviews held prior to steps 7 and 8 are generally concerned with a paper representation of the product rather than the product itself. This approach is required by classical development because a viewable product does not exist until the completion of step 7. For management, true evaluation of status and requirements compliance is either an act of faith or an exercise in massive documentation of the process. In either case, it is not possible for the current status of the product under development to be simply "eye-balled".

The waterfall model dictates that product development will proceed in a strict linear progression. Each design/development/testing step will be performed; the production line will be halted and a formal review will be held. Problems uncovered will be addressed and then the next step will commence. An integral part of each step and its associated review is the production of a significant parallel paper product which supports the review process. Experience over many classical product development efforts has shown that most of these review documents are not maintained in subsequent development steps.

Software Teams

As described earlier, the execution of the serial waterfall steps of classical software engineering is assigned to separate functional disciplines. Each of these functional groups (or individuals) has its own view of the problem and their own idea of the best solution. Such unique views, when transformed into one-of-a-kind solutions, frequently have a significant cost impact on the project. Management assumes a significant task in directing, controlling and enforcing the technical dialog and exchange of information between these functional groups. The required heavy documentation of each step is a manifestation of this technical interface problem.

With a provincial view of the overall process and a technical understanding limited to a single discipline, all processes and products are colored by an insular approach to development. One consequence of direct management concern is quality control. Quality is "inspected in" by an independent control group. Each step review includes formal and informal quality review of the step's work products. The overall level of quality is not determined until the final certification is completed. This is analogous to how American automobile manufacturers used to build their products! Their resurgence in the automobile field is marked by a significant change in the product creation process.

Process Maturity Levels

The Software Engineering Institute's (SEI) Capability Maturity Model (CMM) describes five levels of process maturity. Figure 4 defines the maturity levels in ascending order. The greater the maturity of a software engineering organization, the more effective it is in predictably meeting its technical, quality, cost and schedule goals. Classical software development is generally a Level 1 *ad hoc* process. The uniqueness of each solution and the non-integration of functional disciplines and software management are the prime contributors to this condition. Properly defined, understood and promulgated Domain Engineering and Application Engineering processes and guidelines contribute directly to the maturation and stabilization of higher level capability.

Level		Characteristic	Result
5	Optimizing	Improvement fed back into process	Productivity & Quality
4	Managed	(quantitative) Measured process	
3	Defined	(qualitative) Process defined and Institutionalized	
2	Repeatable	(ad hoc / chaotic)	
1	Initial	(Intuitive) Process dependent on individuals	Risk

Figure 4 - Capability Maturity Model

DOMAIN MANAGEMENT

Domain Management is an activity of Domain Engineering for managing business-area resources to achieve business objectives. Domain Management is an on-going risk assessment of Domain Engineering performance. Domain Management assesses progress of the Domain Engineering effort, assures proper adherence to the domain plans and guides needed revisions to the project strategy and its long term objectives.

Domain Management includes the strategic evaluation of additions to the product-line (domain viability) as well as the identification of resources required to support the expansion. It is paramount that Domain Management coordinate the Domain Engineering activities to support the needs and priorities of targeted product-line applications.

Product-Line Focus

Development and use of domain specific reusable assets is founded on a business case defined by Domain Management. The domain product-line consists of a viable product family which shares sufficient commonality (and predictable variability) to justify an investment in the domain. The business case anticipates cost savings through multiple uses of pre-tested, quality assured assets which can be directly adapted from existing domain assets (code, requirements, design test cases, etc.).

Domain assets are designed to have a significant extended life time. Rather than being developed as unique single purpose entities, domain assets support the product-line for which they were developed. Leveraged reuse of the domain assets across the product-line occurs when an application engineer uses the products and processes developed by Domain Engineering. Leveraged reuse is the adaptation, through variability selection, of components of an existing product-line. Leveraging is one of five progressively effective reuse strategies: *ad hoc*, opportunistic, integrated, leveraged, and anticipated. A high degree of investment return can be expected through leveraged reuse.

Domain Management Products

Because Domain Management is business case driven, its purpose is to facilitate the achievement of the stated business case and the domain objectives associated with it. Strategic planning focuses on determining the nature of the market, identification of business case technical expertise and allocation of available resources between Domain Engineering and Application Engineering. Tactical planning concentrates on the interface between Domain Engineering and Application Engineering by deciding how to apply Domain Engineering resources to create an efficient Application Engineering process which results in the production of high-quality, adaptable products for the instances of the product-line.

Planning, quite naturally, is supported by documented plans. Three types of documentation are used to delineate the Domain Management activities associated with the RSP: (1) Domain Evolution Plan, (2) Domain Increment Plan, and (3) Practices and Procedures.

A Domain Evolution Plan takes a long-range view by defining business models and domain objectives and by organizing resources needed to achieve both. This strategic plan recognizes that not all objectives can be met initially but must develop in the course of time using increments that balance alternative uses of available resources against the potential for return on investment.

Each increment of the Domain Engineering process is planned for and documented by a Domain Increment Plan. This plan is often broken down into multiple Iteration Plans. These tactical plans take the limited view of achieving the near term objectives which define the scope of a specific Domain Engineering increment. Resources and time are balanced to allow creation of enhanced processes and products which directly support the interface between Domain Engineering and Application Engineering.

Domain Management also creates Practices and Procedures documents to codify and coordinate the work efforts of separate groups within and between Domain Engineering and Application Engineering activities. These documents are intended to be reviewed and modified as part of each iteration of domain development.

Domain Engineering Organization

The roles and responsibilities of a two-life cycle organization (Domain and Application Engineering) do not match the traditional single-life cycle acquisition model which most organizations practice. This divergence is a challenge for domain managers during the early adoption phase of Domain Engineering. Additionally, domain managers are also challenged during the adoption phase by a shift to product-line orientation.

The Navy/STARS Demonstration Project lessened these challenges by first limiting the scope of the domain (AVTS) in which the demonstration would occur. This allowed management and the technical team to focus more on the Domain Engineering and Application Engineering processes and less on the product.

A conclusion was drawn that if the process was properly defined and followed then the product was inherent. This limited scope approach also allowed the project to introduce the concepts of product-line development on a smaller, more understandable and less costly level.

Domain Engineering Staffing

Domain Engineering focuses heavily on the process instead of the product! To accomplish this paradigm shift, new engineering and management disciplines are required. Specification, creation and verification of processes necessitates different approaches than the classical functional product-oriented definition and development effort. With the emphasis on adaptable components rather than executable components, significantly different approaches are needed to visualize, create and test the intended domain product.

Whereas the classical approach to software engineering relies on distinct functional disciplines, management of Domain Engineering employs an integrated team approach. The team which is responsible for a given process/product combination participates in all aspects of a spiral iteration cycle. Each of these teams encompasses the expertise required to completely define, implement, measure and evaluate the component to which they are assigned. This front to back responsibility provides the strongest motivation for built-in quality and completeness.

Supplier Integration Into Domain

A domain which encompasses an extensive product-line is likely to include products or tools that are obtained from external suppliers. To the extent that it is possible, these suppliers should become integrated into domain planning activities to promote maximum effective utilization of the products. These products fall into two categories: foundation assets from which domain assets are derived and process support tools which are utilized during the Domain and Application Engineering processes. In either case, the domain assets and associated processes become dependent upon these outside products and their continued availability and quality is an important issue for Domain Management.

RELATIONSHIPS AFFECTING DOMAIN MANAGEMENT

Domain Management concerns for Domain Engineering extend beyond the boundaries of the selected domain and the organization created to engineer and enhance it. The nature of Domain Engineering creates an on-going relationship between the domain manager and the users of the Application Engineering processes created by the domain engineers. A viable domain product-line must effectively deal with these additional domain issues.

Balance Of Power

The concern over software asset reuse extends beyond the DoD. Much of the commercial software development industry is working with various concepts of software asset reuse. As the availability of commercially funded and produced domain assets increases, it may be in the DoD's best interests to consider the use of these existing assets. Utilization of assets developed outside of an organization raises the issues of domain ownership and management. The DoD and commercial entities utilize different business models: the DoD's goal is cost avoidance; commercial developers pursue the goal of maximizing their profit potential.

Today's DoD acquisition model for software production requires that the Government receive Unlimited Rights to the software developed for or used in the corresponding system. As discussed in the first edition of the *DoD Software Reuse Vision and Strategy*, ownership of the rights to the software is a keystone in the motivation for commercial development of domain-specific reusable assets. By insisting on receiving Unlimited Rights, the DoD will potentially stifle the growth and inhibit the supply of reusable assets as well as discourage competition for DoD reuse-based procurements. Adoption of a "Black Box" approach to reusable software wherein the DoD utilizes the adapted assets for their product and maintains Government Purpose Rights for just those assets allows commercial developers to pursue a cost effective profit-based venture.

When a Domain Engineering project involves a team composed of multiple government agencies or a combination of government agencies and commercial entities, management organization and coordination becomes even more critical. The Navy/STARS Demonstration Project is a current example of a team comprised of Government and contractor participants grouped together as Integrated Project Teams (IPTs). Domain Engineering requires an integrated team approach differing from the conventional tiered buyer/supplier approach. It is critical that the primary government agency integrate itself with the other elements of the team so that all members are involved in planning, doing, reviewing and approving. If this relationship is not achieved, management of the domain by the Government becomes unwieldy.

Customer Participation

When the DoD is the owner/manager of a domain which is to be used by various customers, (Government or commercial) it is important that the customer have the capability to provide inputs which affect the planning for, and maintenance of, the domain assets. Within the bounds of the supporting domain architecture(s), a customer should be allowed to provide product requirement specifications for variants which represent new products or processes for the domain.

As a user, the customer must have the capability to provide feedback to Domain Engineering concerning problems or incompatibilities with the adapted assets. Rather than employing customer fixes for these problems, proper Domain Engineering evolution requires that the alterations be performed within the Domain Engineering process by the domain engineers after which a new revision of the domain adaptable assets (domain model) are provided to the customer.

Product Acquisition Lifecycle

Acquisition of domain-based products must undergo a procedural shift in the normal acquisition process. Reuse of domain assets must become a primary goal of the acquisition process. Definition, creation and refinement of the to-be-procured product must be provided for in terms of an adaptable domain model which fits the product-line rather than as a unique set of system functions. This streamlining of the acquisition process must also take into account the contractual aspects of requiring the bidder to utilize assets from one or more domain applications.

The organization, management and statusing of a product-line-based contract must require both the Government and the contractor to embrace and utilize the two-life cycle paradigm of Domain Engineering and Application Engineering with the implied spiral of iterative development cycles. The procurer as well as the contractor must advocate development through reuse of domain assets.

Domain Expansion

Since the assets of a domain persist after they are originally developed, the domain must have a growth plan to ensure its long-term health. There are three natural methods for expansion of a domain product-line.

A domain product-line which supports a number of customers is most easily expanded by responding to customer requests for augmented functionality, increased variability or extended capability. The cyclic nature of the association between Domain Engineering and the customer's Application Engineering activities is a natural growth path for the domain.

As with the Navy/STARS Demonstration Project, most initial domains are actually sub-domains of a larger domain. The AVTS domain is potentially a sub-domain of all Vehicle Training Systems. A second path to domain expansion is to enlarge the scope of a domain to include those elements at the next level which share in the commonality and predictable variability of the existing domain. This technique allows the direct leveraging of existing product-line components without relinquishing any current customer base.

A third domain expansion route selects its direction from an expanded business case analysis. This path is most likely to embrace domains which are parallel to the existing domain. This planned parallelism allows maximum utilization of current domain process knowledge while assimilating new domain technology into the product-line.

CONCLUSION

After two years of direct Domain Management experience with the Navy/STARS Demonstration Project, some important management lessons learned stand out.

Lessons Learned

The most useful lessons learned from any endeavor are frequently those realized from activities which were not actively pursued. These lessons should be noted before beginning a Domain Engineering project:

- Domain Management is a learning process. New management approaches and processes must be assimilated early in the project to have a meaningful affect.
- A Domain Evolution Plan must be maintained by Domain Management to serve as a visible map of the evolutionary trail to be followed. The definition of the domain should be built up incrementally; start small and build increasing complexity with each increment.
- Domain Engineering and Application Engineering are two distinctly different roles. Management must "draw the line" between the two activities but manage the process such that both activities compliment each other.
- Development and management of a viable domain requires that the owners of the domain invest resources and commit to its success.
- Continuity of funding and staffing must be maintained within the Domain Engineering organization to ensure the domain's continued viability.

Recommendation

If an organization is going to attempt a Synthesis-based Domain Engineering project, it should utilize the basic concepts defined by the Reuse-Driven Software Processes. It is recommended that the organization execute a trial pilot project on a small scale and not gloss over the Domain Management activities of this limited test. Place the management emphasis on understanding the dependencies of the Synthesis activities. Lessons learned (management and technical) should be fed back into the process as soon as possible. Above all, concentrate on development of the processes -- if at all possible, do not let yourself become product or schedule driven.

REFERENCES

Defense System Software Development, DoD-STD-2167A, United States Department of Defense, Washington, D.C., February 1988.

DoD Software Reuse Vision and Strategy, 1st Edition, DoD Reuse Program Initiative Office, Falls Church, VA, July 1992.

Paulk, Mark C., Marilyn Bush, Bill Curtis, Mary Beth Chrissis, Suzanna M. Garcia, and Charles V. Weber, *Key Practices of the Capability Maturity Model*, CMU/SEI-91-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1993.

Software Engineering Institute, *Capability Maturity Model for Software*, M.C. Paulk, B. Curtis and M.B. Chrissis, eds., CMU/SEI-91-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1991.

Software Productivity Consortium, *Reuse Adoption Guidebook*, SPC-92051-CMC, Software Productivity Consortium, Herndon, VA, November 1993.

Software Productivity Consortium, *Reuse-Driven Software Processes Guidebook*, SPC-92019-CMC, Software Productivity Consortium, Herndon, VA, November 1993.

SOFTWARE CONFIGURATION MANAGEMENT: A MODERN PERSPECTIVE

John W. Schulke
Staff Scientist
CAE-Link, Binghamton, NY

ABSTRACT

Software development has come a long way in the last decade, but Software Configuration Management (SCM) is just beginning to adapt to the latest technology. Improvements in this area have become key to the success of a number of new initiatives, including Reusable Software and System Concurrency. SCM should not be a burden to engineers. It should not be an overhead to managers. In fact, SCM, done well, can be one of the most significant cost savings and avoidance factors available to industry today. Presented in this paper is a refreshing look at SCM as a vehicle for improvements in the software engineering process and philosophy of control.

The increased complexity of modern training devices and advances in the technology of the development environment have dramatically increased the size and complexity of the SCM problem, thus demanding more discipline and control. To meet this demand, SCM must be inherent in the very way business is done. It must be owned by the entire development team and be supported by efficient tools that enforce the process. This control must include every aspect of the development process. The term used for this discipline is "self-governance".

Applied correctly, self-governance will result in significant cost savings and process improvements. Pay-backs result from improved maintainability, reductions in process cycle time, and the ability to properly support reuse. This paper advocates a phased approach to SCM that fits naturally into the engineering process. The right amount of control is placed into the hands of the people who are best able to accomplish the required tasks during each project phase. By having SCM tasks, such as software release and change control, performed as a simple part of each software developers day to day activities, the costly "crisis events" that tend to occur due to loss of control can be prevented.

The information age is upon us. Future advances will increase the demand for discipline and control. The concepts presented in this paper are simple and the potential pay-back is great. The challenge is to implement self-governance effectively in order to meet the technical challenges of the future.

About the Author

Mr. Schulke is a Staff Scientist with CAE-Link Corporation in Binghamton, NY. In his current position, he is responsible for Computer Program System Management tools and procedures on the System Support Center of the B-2 Air Crew Training Device. He received his BA Degree in Electrical Engineering from Tri-State University in 1964 and has been employed by Link for over 25 years.

SOFTWARE CONFIGURATION MANAGEMENT: A MODERN PERSPECTIVE

John W. Schulke
Staff Scientist
CAE Link
Binghamton, NY

INTRODUCTION

This is a time of rapid change in the software development business. Our whole economy is lean, resulting in massive cutbacks in industry. The increased competition, both domestic and foreign, has motivated corporations to look hard at internal software processes, methods and tools. American companies are streamlining to meet the demands of stiff competition from highly-tuned software development teams. Never before have we seen such an emphasis on productivity and quality. Yet, the cost of software has continued to rise despite new languages, methodologies and tools.

In this paper we will discuss how Software Configuration Management (SCM) plays a key role in the quest for quality and productivity in the software development process.

Consider the following factors that weigh heavily on the cost of software.

THE IMPACT OF DEFECTS

In "Software Quality and Testing: What DoD Can Learn From Commercial Practices", Lieutenant Colonel Mark R. Kindl (U.S. Army) states "Early identification and correction of errors are critical to software product correctness and quality. Correcting errors in software is a fix, but not a solution. Software errors are often the symptoms of a more fundamental process defect."

A method of recording and tracking problems is mandatory in order to solve basic process problems. So often tracking of errors does not even begin until the later phases of a project. The biggest mistakes in software development are almost always made during requirements definition and design. Data from IBM Federal Systems Company (FSC) in Houston shows that the average error repair cost increases 10 times in each successive phase of the lifecycle in which the error is detected.

Where does all this lead? Ltc. Mark R. Kindl summed up the reason that one organization is able to produce higher quality software than other organizations. He states: "The difference results from a strong attitude toward quality, the disciplined practice of its basic processes, and a commitment to process improvement. From manager to programmer, the entire organization strives to achieve zero defects through prevention." Though this success is undoubtedly the result of many factors, we contend that a well-disciplined SCM approach is a major contributing factor.

THE IMPACT OF THE DEVELOPMENT PROCESS

In "Decline and Fall of the American Programmer", Ed Yourdon asks the questions: "What do world-class software organizations do differently from your organization? What tools do they use? What methods, procedures and techniques do they use". No "silver-bullet" answers are provided, but clearly Mr. Yourdon's research shows that world-class organizations share a commitment to a number of key software technologies:

1. CASE (Computer Aided Software Engineering) technology
2. Metrics
3. Object-Oriented Methods
4 Software Reuse
5 Continuous Process Improvement
6 Training

We can sum these up to be "a commitment to process improvement and methodology". The stability and predictability of the development process used within a given organization is a key factor in the cost of software. Even a poor process that is stable and faithfully followed by the software organization is likely to out-perform the most modern methodology that is not well documented, controlled and understood. The cost of a paradigm shift in an organization can be tremendous. All too often failure can be attributed to poor management visibility and control, or to the inability of management to react to changing times. Managers at all levels must understand and enforce whatever process is being used by their organization. Change should be gradual and well planned.

Lessons on Process Improvement

W. Edwards Deming was the genius who is often credited with revitalizing the Japanese industry. In the book "The Deming Management Method", Mary Walton presents methods which helped change the Japanese industry and which can do the same in America. Mr. Deming's emphasis was on quality and productivity. Many of his principles apply directly to the software development process and to SCM. Walton asks the question: "How do you improve quality and productivity". Most people would say, "By everyone doing his best". Per Walton, this is not correct. She says, "The system is such that almost nobody can do his best. You have to know what to do, then do your best". Therefore, the first step is to have a plan. Second, document the plan, and then teach it to everyone and continue to teach it. The plan ultimately takes the form of a process. This process will require continual improvement.

The Deming methods require the adoption of a new philosophy. Americans are too tolerant of poor workmanship and sullen service. Mary Walton states, "We need a new religion in which mistakes and negativism are unacceptable. Quality comes not from inspection, but from improvement of the process. With instruction, workers can be enlisted in the improvement." This is where the real benefits from SCM can be realized.

The Process Maturity Model

In "Process Improvement and the Corporate Balance Sheet," Raymond Dion states: "A key requirement for the success of a new software development process is the accurate evaluation of how effective it is in reducing the bottom line cost of getting the job done". Mr. Dion further describes how the Raytheon Corporation was able to successfully measure the effect of process improvements by following the principles of W. Edwards Deming and Joseph Juran. Basically, their philosophy teaches that real process improvement must follow a sequence of steps, starting with making the process visible, then repeatable, and then measurable. SCM is the natural vehicle to institute this type of measurable process improvement. Its basic philosophy advocates control, stability, and feedback.

In his book "Managing the Software Process", Watts S. Humphrey states: "When asked to name their key problems, few software professionals even mention technology. Their major concerns are open ended requirements, uncontrolled change, arbitrary schedules, insufficient test time, inadequate training and unmanaged system standards". These are all management and control problems. Mr. Humphrey's book details the concepts behind the Software Maturity Model which is being used today to evaluate an organization's software development capability. A five-level maturity framework was developed at the Software Engineering Institute (SEI) of Carnegie Mellon University. All software organizations fall within one of these five levels. The ability to move to a higher level is accomplished through software process improvements.

Mr. Humphrey further states: "The software process is the set of tools, methods, and practices we use to produce a software product. The objectives of software process management are to produce products according to plan while simultaneously improving the organization's capability to produce better products. The basic principles are those of statistical process control, which have been used successfully in many fields." Control and visibility are essential to the success of any process improvement initiative. The remainder of this paper will outline a concept of SCM that is designed to help transform a software organization into a productive, quality-minded team.

THE CONCEPT OF "SELF-GOVERNANCE"

We refer to the proposed concept of SCM as "Self-Governance". Self-governance can be defined as the ability of an organization to govern itself from within at all levels. Such an organization will have a carefully defined and documented software development process. The operation will tend to be self reporting and self correcting, with a quality mind-set being promoted throughout the organization. To a self governing-organization, SCM is a necessary part of their plan for success. Self-governance incorporates many of the functions of Configuration Management (CM) and Quality Assurance (QA) organizations into a team approach at the engineering level. This does not eliminate the need for independent CM or QA organizations, but instead changes their function to one of process (not product) inspection and control. The intent is to monitor the quality of the processes by which software is specified and developed, and to provide the feedback necessary for self-sustained improvement in the process.

How does Self-Governance and SCM fit into the plan for success? It is the mechanism for visibility, control and feedback. DoD Standard 480A provides the following definitions:

> Configuration Management: A discipline applying technical and administrative direction and surveillance to (a) identify and document a Configuration Item (CI), (b) control changes to those characteristics, and (c) record and report on change processing and implementation status.
>
> Configuration Control: Systematic evaluation, coordination, approval/disapproval and implementation of all approved changes in the configuration of a CI.

Although these definitions generally apply at a much higher level than is intended for purposes of this paper, they adequately describe the concept. However, the way that SCM is often applied is far from what is necessary to help us improve our quality and productivity. In fact, SCM is usually applied too late and improperly to help us analyze and improve our software development process. And, it is often applied for all the wrong reasons.

A Mission Statement for SCM

Let us start by defining the goals of a Self-Governing SCM system in the form of a mission statement:

> SCM shall be practiced as an integral part of all software development activities in order to:

- o Provide early baseline definition/visibility
- o Enforce a phased development process
- o Provide readiness status for each phase
- o Control and track problems to their solutions
- o Maintain feedback in the form of metrics
- o Correlate bid statistics against actuals
- o Promote the reuse of software components

Characteristics of Self-Governance

Now let us examine the characteristics of a Self-Governing Organization and then look at how SCM is essential to its implementation and operation:

1. Accountability:

   The successful software development organization recognizes that quality and productivity must be owned at the grass roots level, and must be promoted from the top levels of management. The development process will be understood and adhered to by the entire organization. Each person holds quality as a major part of his/her job. Training is considered essential for success.

2. Self-improving processes:

   Successful organizations always have a well-defined software development process. The existence of a quality-minded development process is far more important than the specific methodology or life cycle model. The ability of the organization to follow, improve, and maintain the process will determine the ultimate success of the organization. The goal is to detect errors early to avoid the high cost of changes found in the later phases of a project. A concept of no-fault error reporting is encouraged so that errors are willingly reported and corrected. Metrics are gathered and analyzed to facilitate process improvement, not to place blame on any individual or group. This is vital to maintaining a self-correcting process.

3. Team concept:

   The most successful organizations have always been those that promote development of small, highly-integrated teams. There are many ways that this can be accomplished, but, whatever the approach, an environment that promotes good communication, ownership, quality, motivation, and trust is essential. Good documentation and control is a natural fallout of the process by which the teams operate. The development process fully defines the operation and the team provides a majority of its own training and process correction. Interfaces with other teams must be particularly well managed and documented.

   Teamwork is essential to the Deming Methods. "We, in America, do not have it," says Mary Walton. Software companies must decide how they will institute teamwork into their organizations. There must be a consistency of purpose within the teams and across the teams. Quality first with a win/win attitude will do more to motivate people than any form of rigid quotas that take account of statistics, but not of quality or method.

4. Generation of Metrics and reports:

   Organizations that are able to improve their quality and productivity will place a high value on metrics. The SCM system is a natural source of much of the information necessary for schedule evaluation and process improvement.

Critical Business Factors

In addition to those characteristics of a self-governing organization, some key initiatives are necessary to survive in today's market place:

1. Promotion of Reuse:

   Reuse is a planned and controlled methodology by which Reusable Objects (RO) are designed, preserved and maintained with a goal of high repeated use across multiple disciplines and projects. If applied properly, it provides one of the most significant cost reduction opportunities available today. Key to its success is a well-controlled, easily-accessed software library and a reuse methodology.

2. Promotion of phased integration:

   Successful software development organizations are able to optimize schedule. Large projects such as aircraft simulation devices require a complex integration plan for hardware and software. A phased integration process, with a focus on reducing integration complexity, bottlenecks and schedule, provides the potential to reduce schedules by as much as one third.

3. Support of Parallel Development and Concurrency:

   Rapidly changing requirements and concurrency issues are a way of life today. For an organization to survive, its software development plan and SCM system must deal with these costly items.

SCM AND THE SELF-GOVERNING ORGANIZATION

Figure 1 illustrates the benefits of a well-implemented Software Configuration Management system:

1. Productivity is increased by having a well-organized and planned development process and by the support of reuse.

2. Process improvement results from the stability of an enforced process and the feedback resulting from the capture of statistical data about the process.

3. Management visibility is greatly increased by the existence of up-to-the-minute status information and the ability to report on baselines and concurrency data.

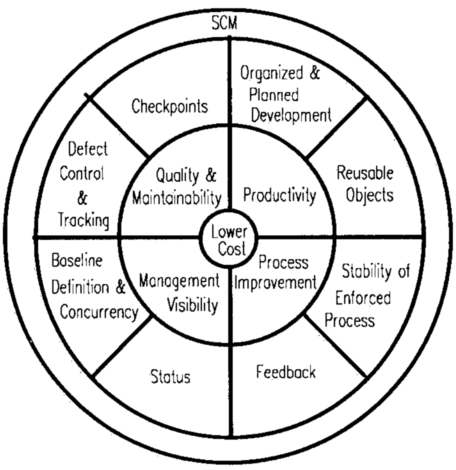


Figure 1 SCM Benefit Diagram

4. Quality and maintainability are ensured by enforced checkpoints within the process along with the existence of a defect control and tracking system. The end result is a lower cost of software development.

It must be emphasized that these savings are not automatic. Knowledge and care are required to implement an efficient and effective SCM system. Let us take a more detailed look at SCM.

SCM Capability Recommendations

SCM methods and tools can very considerably across organizations; however, the results should be the same. SCM should be built into the process by which business is done. It should not be optional or easily misunderstood. It should be user friendly and natural. The entire organization should believe it is important. If any part of the process is unnecessary, or is viewed as unnecessary, it will not happen or will produce meaningless results. This is a culture issue. Training and teamwork are a must. All too often, software developers do not understand the necessity of a controlled environment.

The process must follow the controlled sequence of events: identification, release, and change control. The data captured for each Configuration Unit, CSC, and hierarchical node must be necessary, and should result in reports that contain all metrics necessary for schedule evaluation and process improvement. Wherever possible, the data should be automatically calculated and entered. Unambiguous entries should be enforced or be automatically selectable. The major SCM events are discussed below.

1. Identification: This is the process of identifying the parts of the software hierarchy starting with the top level structure and CSC organization, and ending with a complete "Family Tree" or hierarchy containing all configuration units allocated to their specific node in the tree. This is extremely important to allow the project to size and schedule its tasks. A preliminary tree will most likely be created as part of the proposal. Much of the visibility and feedback comes from this process.

2. Informal Control: This level of control begins as early as possible and is the responsibility of individuals and teams. Each configuration unit should be processed into a Software Control Library soon after they are identified. Preferably, this will be done electronically with some form of integrity evaluation and unit history recording. Changes should not require formal approval at this level, but a measure of the level of activity should be provided. Simplicity and ease of operation are a must. Users and teams should want to use the control mechanism because it helps them organize and perform their tasks. However, team leaders must ensure that this happens. A tool may be helpful. The teams should find the libraries indispensable for sharing of information, and the project will appreciate the availability of status and metrics.

3. Release: The first release of configuration units is the beginning of formal control. This should begin when there is a significant change in the criticality of the specific configuration unit or group of units. For example, source units should be released prior to CSC integration. The entire CSC should be released at one time and a revision number should be assigned to each configuration unit (source file).

Each change to a released item should be in response to a specific problem, and tracked by a change authorization process and a Software Problem Report (SPR). All configuration units that are modified to fix a problem should be identified in this SPR so that reports produced can be used to correlate the problem with a type and size of the change. An analysis of each problem should be performed and the defect category should be captured. All change activity must be related to an authorization account such as an ECP and approved before work is allowed. This is necessary to insure that all changes are required and budgeted. Work must be constrained to the baselined requirements.

Software Control Library (SCL)

A well-defined and implemented Software Control Library is necessary to properly manage a software development project. It is the heart of the SCM system. It must be a controlled environment in which all configuration units are maintained and administered. The history of each configuration unit should be automatically updated each time a change is made. This can be done using existing control tools, such as SCCS under UNIX or CMS under VMS. The SDL is generally organized functionally with the ability to easily break out individual CSCs and provide changing levels of control for each. More sophisticated implementations will provide verification features and a test environment. For example, source units may be precompiled before they are allowed into the library. A review process will be required prior to release of units or changes to released units.

The SCL should provide the ability to increase the level of ownership and control of individual components (CSCs) as they progress from one development phase to the next. However, at all levels, the library should be open for review and design interchange at all times. It should be easy to generate reports from the library and any associated database.

To be fully controlled, all processes and tools that operate on the SDL must be controlled. This includes the ability to automatically process the configuration units into loads, and may include the ability to integrate many CSCs into integrated loads. This load building process should perform the following as a minimum:

1. Select units to be processed into load libraries.
2. Distinguish between versions (Variants) of units which apply to specific loads.
3. Control the compilation and linking of units into tasks.
4. Collect tasks and data files from load libraries onto distribution media.
5. Control the distribution media, and associated documentation.

REUSE CONSIDERATIONS

Reuse is another productivity methodology that requires a well-conceived SCM environment. Both industry and government organizations have embraced reuse as a key initiative for the 1990s and beyond. Various forms of reuse have been used from the very beginning of the software industry. Most reuse of the past can be classified as ad-hoc, or user dependent. This is not the form of reuse which is envisioned for the future. To provide a significant cost advantage, reuse must accomplish the following:

1. The object being reused should be large enough to significantly reduce documentation, test, and integration time. This does not eliminate the possible benefits of ad-hoc reuse or of the reuse of smaller components, but emphasizes the benefits of larger Reusable Objects (ROs).

2. A reuse repository must exist which provides easy access to the ROs. Included with each RO should be a description file or database which fully defines the RO for easy identification and selection by query. ROs should be functionally grouped within the reuse library for ease of access and searching. A mechanism of certification should be provided in order to establish preferred products.

3. The selection of ROs for a project should be accomplished as early as possible. In fact, it is best accomplished at the time the software is bid and proposed, thus reflecting a reduced development cost and schedule. At the start of a project it must be easy to transfer ROs from the reuse repository to the project. The project should accept these ROs as released components and they should be ready for integrated testing as soon as possible without significant rework or documentation effort. Changes made on the project, however, should be controlled like any other software change.

4. A standard RO design methodology should be established within a software development facility. The enforcement of this methodology should be rigid enough to enforce the integrity of the library system, yet, open enough to allow functional groups to make decisions based on the unique characteristics of the specific software being implemented. For significant reuse it is necessary for the entire organization to develop and adhere to certain architectural and interface standards, taking advantage of open system standards.

CONCURRENCY CONSIDERATIONS

In the training device market concurrency management is a necessity today. Simulation devices such as the B-2 WST are often developed in parallel with the vehicle being simulated. Almost always, multiple configurations of a vehicle are in development simultaneously. Without a disciplined control system, this becomes extremely costly or impossible.

Baselining is an important concept with respect to concurrency. DoD Standard 480A defines three baselines: 1). Functional Baseline, 2). Allocated Baseline, and 3). Product Baseline. These three are, in general, distinguished by the type of items controlled and the milestones at which they are established and approved. These are contractual baselines which must be addressed by most government contracts. However, for purposes of this paper we will extend the concept of baselining to include informal baselines. These will provide the ability to distinguish between multiple applications of a product that may be developed in parallel. An example would be an aircrew training device having multiple deliveries in parallel, with each delivery representing a different variation of the aircraft.

Software must be baselined prior to formal acceptance of a device. It is usually verified and approved by a government Physical Configuration Audit (PCA) and a Functional Configuration Audit (FCA). Internally, however, the baseline process must begin much earlier and the SCM process must provide a means to distinguish software that is common between devices and software that is unique to a specific device and, therefore, a specific baseline. To accomplish this, the change process must acknowledge variants of configuration units and a load build process that will produce unique loads for each device. Without a well-conceived SCM process, this type of control would be a nightmare or perhaps impossible. Changes at this level are very formal and require a means to isolate changes to a specific baseline and to report on change activity that relates to a specific baseline.

Baselining and concurrency issues are illustrated in the following case history.

A CASE HISTORY - B-2 WEAPON SYSTEM TRAINER

The B-2 Weapon System Trainer (WST) program will be used as a case history of a self-governing organization. The contract consisted of six WSTs and two Mission Trainers (MTs). The development of these training devices provide some interesting lessons for evaluations. The contract was awarded in early 1985 and was CAE-Link's first Ada contract.

Concurrency of the WST with the B-2 aircraft was a primary requirement. At times, two and even three baselines were in various stages of development and use at the same time. Each baseline represented a different configuration of the aircraft. Due to the size of the development effort, SCM was a critical factor in the ability to maintain schedule and concurrency. Over 38,000 software configuration units were managed by the System Support Center (SSC). These units can be broken down as shown in Table 1. The large size of the project can be better visualized by the Ada Lines Of Code (LOC) breakout shown in Table 2.

As the result of the rigorous testing and careful tracking of software problems, the number of Test Discrepancies (TDs) encountered during customer acceptance was low for the size of the device. For example, there were only 483 software test discrepancies on the first device delivered. Early detection of problems is evident from the number of problems isolated early in the development process. Table 3 illustrates the distribution of these problems. Note that 25 percent of the changes were related to funded design changes. Another 25 percent were related to coding errors which are the least costly to fix. Very few were caused by requirements allocation problems. The design and interface errors (25% and 5%) are the most costly to fix. Most of these were found during the code and test phase. Analysis of these design problems indicates that many were caused by changing aircraft data and other concurrency-related issues.

A majority of the support software, except for operating systems and compilers, was developed specifically for the SSC under this contract. This was necessary because very little Ada support software existed at the start of the contract. In fact, the entire Computer Program System Management (CPSM) component was developed under the contract. CPSM includes the SCM and load support tools. The software development process was defined and in place before the tools were implemented.

Table 1
Software Configuration Units

Category	SSC	WST/MT	Common	Total
Documentation	260	751	193	1212
Software				
Non-Variant	8084	14584	4751	27419
Variant	97	7444	491	8032
Firmware				
Non-Variant	0	1401	130	1531
Variant	0	20	0	20
				38214

Table 2
Lines of Code

Major Elements	Ada LOC	Non-Ada LOC	
WST	483,950	579,400	
Common (WST/SSC)	105,117	5,100	
SSC	434,585	130,700	
Total	1,023,652	715,200	1,783,852

Note: Non-Ada includes Job Control Language, Scripts, data and other languages

Table 3
Software Change Activity Following Release (WST/SSC)

Change Category	# of Problems	Percentage of Problems	Units Changed
Design Error	1916	24.18	6611
Interface Errors	431	5.44	1046
Coding Errors	2001	25.26	3704
Documentation Errors	559	7.06	250
Misc. Document Errors	909	11.47	683
Requirements Allocation	56	.71	331
Pilot Tailoring	24	.30	43
Design Changes	2027	25.58	5251
	7923	100.00	17919*

* A single unit may have been changed multiple times through the change activity phases.

Self-Governance and the B-2 WST Organization

In terms of control, the SCM process was quite successful on the B-2 WST as witnessed by the number of units released and the number of changes controlled. The Computer Program Development Plan (CPDP) and the Standards and Procedures manual completely defined the software development process.

Self-Governance was introduced to ensure quality at the engineering level. Quality checkpoints, such as internal reviews and code walk throughs, were carefully followed. Engineering assumed the primary change approval process on the Software Configuration Control Board (SCCB). The SCM tool was enhanced to automate many quality checks. The Quality Assurance organization closely monitored the processes being performed. The

Configuration Management organization became the owner of all released software, but Engineering authorized and administered all change activities.

The development process was continually reviewed and improved over time. The number of design errors was a subject of much concern. The review process was improved and enhanced by computer-supported quality checks. Prologs in source code were automated using data available in the SCM database.

The Software Change Process

Software configuration units were identified very early in the development process. Components were allowed to proceed in development at their own rate and early integration was encouraged. Prior to integration, however, all units in the CSC were required to be released and placed under formal change control. A single integration path was followed. This forced all components to be processed into a single simulator test load for each baseline. The load was built on a daily basis for use by the test team. Extensive testing was also accomplished on the SSC.

Figure 2 shows the software change processing cycle. A system or software problem could be identified by anyone, but formal test discrepancies were written up by QA and the customer. Informal discrepancies were usually identified by Engineering and problem reports were written on each problem by the engineer. This was done through an on-line SCM system and the information was recorded on a Change Authorization Notice (CAN). The resulting Software Problem Report (SPR) displayed the following information: problem description, priority, classification, and authorizing activity (ECP). Certification of the problem was required by the functional team lead before the CAN was sent to the responsible engineer for analysis. The responsible engineer recorded his/her analysis on the CAN, again using the on-line SCM system. This provided the following information for the Software Problem Analysis (SPA) report: recommended solution, units affected, baseline effectively, pre and co-requisite changes, problem category, and cost estimate. Approval was required by the functional team lead before work was authorized to begin.

An engineer was assigned the responsibility to make the correction. Units to be changed were identified on line through a Computer Program Change Request (CPCR) form. A copy of these units were brought forward to the engineer's workspace from the SCL so that they could be edited and tested. Before the changes were allowed into the test load they were precompiled into sublibraries by an analysis tool to prevent breakage of the master libraries and to determine the compilation effectivity. The change was then applied to the daily load and verified. Several passes might be required to get the change working properly in the engineering load. The CAN and CPCRs would then be closed, following an approval cycle by the SCCB. Only then would the change be applied to the Qualification load which was used by QA and our customer for final verification. Through the entire process, the SCM software maintained status and recorded dates for all events. Status was readily available through reports and on-line queries.

Concurrency Issues

A Load Processing Diagram is shown in Figure 3. Note the existence of a single set of Software Control Libraries (SCL). These were owned and controlled by CM. All released source resided there, along with change history records under DEC's Code Management System. In addition the SCM tool maintained a separate database of CM information on all configuration units and change activity. Variants of the configuration units were used to track changes against each baseline. Separate baselines were used to control independent loads and, as can be seen in the figure, a set of development libraries was associated with each baseline. All software related to a particular baseline was automatically selected from the SCL for processing into the set of development libraries associated with that baseline.

The SCM system was able to distinguish between "common" units and units that were unique for each baseline. In this way a baseline could be easily "cold started" from source at any time. Each new baseline started as a mirror image of a previous baseline. The system allowed changes to be isolated to specific baselines or applied to multiple baselines in parallel. This was accomplished by applying changes to a parent baseline first and then "block updating" selected changes to child baselines. Parent library units were never allowed to inherited from children. The SCM software forced all changes to be initially applied to a single baseline by an authorization mechanization which used ECP numbers for control.

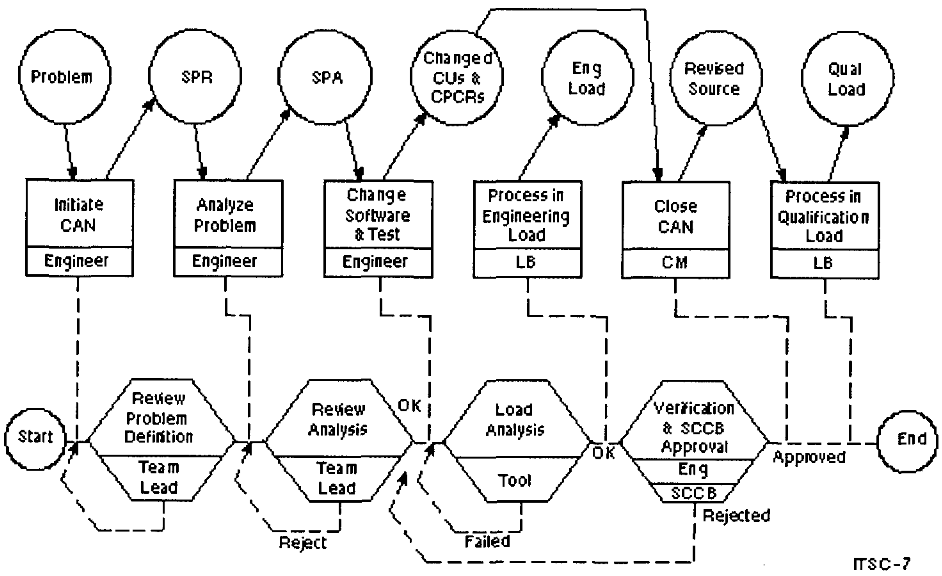


Figure 2 Change Processing Cycle

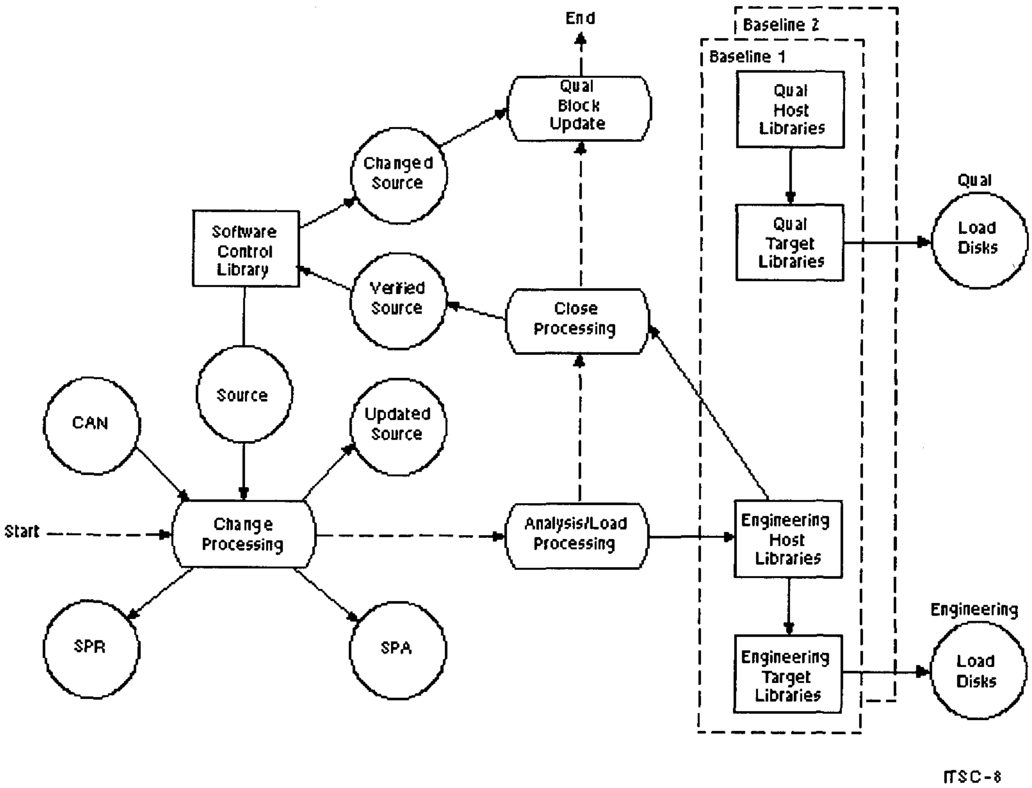


Figure 3 Load Processing Diagram

The Load Building Process

All software development was initiated on a host DEC computer complex. Unit testing and component verification was accomplished there. An automatic load build process managed the processing of changes and new releases into a single load which was distributed across multiple target computers in preparation for real-time integration. An individual load consisted of more than 13,000 Ada source units which had to be compiled into both host and target libraries. A breakout of all configuration units for a load is shown in Table 4. The host Ada libraries were used to select the compilation effectivity for the tasks assigned to each target library. The configuration control process managed the distribution of data files to the proper target libraries and load disks. All processes and load processing software was controlled in order to allow ease of maintenance and to provide the cold start capability. Because of the large number of Ada units involved, loads were built overnight. Cold starts were seldom required, but when they were it took three to five days to rebuild a load from scratch.

Table 4
WST/MT Configuration Unit Breakdown

Functional Area	Ada Units	Data Units	
Common	1842	496	
Air Vehicle	2348	253	
Avionics	916	163	
Digital Radar	2811	3665	
Instructional Systems	560	5021	
Computer Systems/Other	4893	6021	
	13370	10970	24340

Lessons Learned

Because the B-2 WST was our first Ada training device we learned many SCM lessons.

1. Unused data entries – It is extremely important to plan software metrics in advance. We had an idea what information was necessary to manage the project, but some parameters were introduced because it was thought they might be important later. This forced users to enter information that was never used. On the other hand, we found data that was required and not provided. This forced database changes and automatic filling of data fields. Careful planning and research is necessary before defining a company's SCM requirements.

2. Tightly coupled components – Early inexperience with Ada in the large caused components to be too tightly coupled. This forced changes to be made at levels that affected multiple components too late in the schedule. It also forced most components to be integrated directly into formal loads instead of allowing independent component integration.

3. Multiple target computers – The complexity of the load building processed was greatly increased by the number of different target computers used. In the case of the B-2 WST, this was caused by the limited availability of Ada compilation systems in the early days of the project.

4. Formal control too early – Inadequate provision was made for informal control of configuration units during early development phases. This forced a "big bang" release of units just prior to component integration. Informal identification and control is recommended prior to component integration to reduce unnecessary constraints on software developers.

5. Unnecessary reviews – Some reviews in the change control process proved to be unnecessary as self-governance became effective. The team leads were allowed to use them or not use them as required. In many cases, an automated check in the SCM tool provided adequate integrity checks and eliminated the slow human review process.

RECOMMENDATIONS

In this paper we have shown how cost, quality and productivity are highly dependent on a company's SCM implementation. Self-Governance is a concept of SCM which deals with the issues of process improvement, quality, reuse and concurrency. There is a message here for every software development organization. Plan to make quality and control a part of the mindset of every member of your organization. Do not expect SCM to solve your software development problems. It is a vehicle by which an organization can enforce and improve its development process through control and visibility. Use it wisely and it will be of great benefit to your organization. Understand that it will take planning, commitment and time.

REFERENCES

Barnard, J., "Managing Code Inspection Information", IEEE Software, March 1994, page 59.

Bersoff, E.H., Davis, A.M., "Impact of Life Cycle Models on Software", Communications of the ACM, August 1991/Vol. 34, No. 8.

Dion, R., "Process Improvement and the Corporate Balance Sheet", Software Engineering Technology: CrossTalk, February 1994, page 7.

Hegde, S.S., "Introducing a Configuration Management Solution Corporate-wide - An Experience Report", IEEE Software, June 1992.

Humphrey, W.S., "Managing the Software Process", Addison-Wesley Publishing Company, Inc.

Kindl, M.R., "Software Quality and Testing: What DoD Can Learn from Commercial Practices", Software Engineering Technology: CrossTalk, December 1993, page 10

Linger, R.C., "Cleanroom Process Model", IEEE Software, March 1994, page 50.

Walton, M., "The Deming Management Method", The Putnam Publishing Group, New York, NY, 1986.

Yourdon, E., "Decline and Fall of the American Programmer", PTR Prentice Hall, Inc., 1993.

INTRODUCTION TO THE INTERNET

Dr. Ann E. Barron
Associate Professor
University of South Florida
Tampa, Florida 33620

ABSTRACT

The Internet is a worldwide telecommunications system that provides connectivity for thousands of other, smaller networks. The "backbone" for the Internet consists of high-speed, long-distance data lines that were built by the National Science Foundation in the 1980's.

No one owns the Internet; the costs of operations are shared jointly by its users: educational organizations, government research agencies, the military, and private organizations. Several sources estimate that as many as 30 million people may be connected to the Internet and that the Internet is growing at a staggering rate of over ten percent per month.

The benefits of the Internet for industry and military are enormous -- information can be located in international databases; up-to-the-minute weather data, economic information, and images can be obtained; messages can be sent throughout the world in a matter of seconds, and huge electronic files can be transferred quickly and cost-effectively. This presentation will provide an introduction and overview of the Internet and its applications in industrial training.

ABOUT THE AUTHOR

Dr. Ann E. Barron is an Associate Professor in Instructional Technology at the University of South Florida where she teaches graduate level courses in multimedia and instructional design. She is an internationally recognized leader and presenter in the field of interactive training, the co-author of a recent book, *Multimedia Technologies for Training: An Introduction*, and the Executive Editor of the Journal of Interactive Instruction Development.

Dr. Barron is also the Associate Director of the Florida Center for Instructional Technology. In this role, she has designed and developed several instructional simulations, booklets, and brochures for the educators of Florida to acquaint them with telecommunications and the Internet. She may be contacted at the University of South Florida, EDU208B, Tampa, Florida 33620; telephone 813-974-1631; Internet BarronA@mail.firn.edu.

INTRODUCTION TO THE INTERNET

Dr. Ann E. Barron
Associate Professor
University of South Florida
Tampa, Florida 33620

INTRODUCTION

The Internet is a worldwide telecommunications system that provides connectivity for thousands of other, smaller networks; it is often referred to as a network of networks. The Internet originated in 1969 when the Department of Defense funded a network called the Advanced Research Project Agency (ARPANET). This network was designed primarily for military applications.

The ARPANET was expanded by the National Science Foundation in the 1980's when they built a backbone of high-speed, long-distance data lines across the nation. This backbone is now referred to as the Internet; it acts as a conduit to transport electronic data from one network to another network on a global basis.

It is difficult to measure the number of computers on the Internet because so many computers are connected to networks — that are connected to other networks — that are connected to the Internet. One source estimates that as many as 30 million people may be connected to the Internet either directly or indirectly (Kent, 1994). Another source reports that "the Internet is growing at a staggering rate of over ten percent per month" (Dyrli, 1993, p. 54).

Through the Internet, information can be located in international database; up-to-the-minute weather data, economic information, and images can be obtained; and messages can be sent throughout the world in a matter of seconds. The connections available through the Internet offer such a vast amount of information that companies all over the world are seeking ways to obtain access. In fact, "two new accounts are added every four minutes, and one of these accounts is from a commercial site" (Thorell, 1994, p. 53).

There are several ways that industrial trainers can use the Internet connection to communicate and obtain information: (a) They can collaborate and send messages through electronic mail, (b) they can access databases of information located on computers around the world, and (c) they can transfer files electronically.

ELECTRONIC MAIL

A major benefit of an Internet connection is that electronic mail (E-mail) messages can be exchanged on a worldwide basis—people in the United States can communicate directly with people in Germany, China, and numerous other countries (Barron, Ivers, Hoffman, & Sherry, 1994).

Internet E-mail is especially advantageous for business correspondence because it is inexpensive, fast, and it can be sent at any time. The time differential between continents is not as important with E-mail because even if it is the middle of the night on the other side of the world, the message will wait for the recipient to check for messages the next day. The Internet is making telephones and fax machines far less essential, and it is providing financial savings for many companies (Ubois, 1994).

The interface for sending and receiving E-mail messages varies based on the computer, software, and network system. Most of the interfaces are menu-based and quite easy to use. For example, PINE is a very popular E-mail interface. It is public domain software and offers the features of direct answers, spell-checking, and automatic forwarding of messages.

In order to reach the intended recipient in a timely manner, Internet addresses must be very specific. For example, an Internet address may look like this:

barron@madonna.coedu.usf.edu

In this case, *barron* is the name of the person; *madonna* is the computer her account is on; *coedu* is the building (College of Education) in which the computer is located; *usf* is the educational institution (University of South Florida) and *edu* denotes an educational organization.

Internet accounts for other countries and organizations will look slightly different. An Internet address for the United Kingdom may end in *uk*, instead of edu; an Internet address in the government will end in *gov*; the military is *mil*; and companies are usually *com*. For example, the President's address is *president@whitehouse.gov*.

E-mail is often used in the distribution of training and education. For example, the University of South Florida offers several graduate level courses via Internet E-mail. Students can communicate with their professors and peers on a continual basis, without being constrained by geographical locations.

An extension of E-mail, called newsgroups, also provide electronic means for collaboration and research. Electronic newsgroups are similar to bulletin boards or forums in which people can leave messages for others, ask questions, and respond to inquiries. There are currently over 5,000 Usenet newsgroups that provide means for the distribution and discussion of the latest developments in technology and training.

REMOTE ACCESS

In addition to direct E-mail communications, many trainers use Internet connections to access information on distant databases. A major benefit of the Internet is the ability to connect a computer directly into another computer system at a remote location. For example, you can log into a distant computer, read the files, search a database, or leave a message for others to read. This ability to use another computer that is connected through the Internet is called remote access or "Telnet."

There are several thousand systems with Telnet access on the Internet. For example, you can telnet to NASA Spacelink in Alabama, read the documents and download graphics or text. For this, and other systems, you must know the telnet address and the sign-on procedure. Examples of telnet sites that are useful to industry and military environments include:

Federal Register. Up-to-date regulatory data and notices from all U.S. government agencies. Also includes the Commerce Business Daily. (gopher.counterpoint.com)

Foreign Exchange Rates. Daily updates on the exchange rates from the Federal Reserve Bank in New York. (gopher: una.hh.lib.umich.edu)

National Technology Transfer Center. Includes information on federally sponsored research efforts. (iron.nttc.edu)

InterBEX(Business Exchange). A free, commercial information service for individuals and vendors. (E-mail: interbex-index@intnet.bc.ca.)

Electronic Newstand. Selected articles from over 74 magazines. (Gopher: gopher.internet.com)

Economic Bulletin Board at the University of Michigan. Contains files from the U.S. Department of Commerce's Economic Bulletin Board, including information on foreign trade, industry statistics, and economic conditions (Calcari, 1994).

FILE TRANSFERS

The File Transfer Protocol (FTP) was developed in order to facilitate the transfer of files on the Internet. With FTP software, users can access files, examine directories, and move programs to or from a remote computer.

The most common method for transferring files is called anonymous FTP. For example, if you wanted to obtain a file that was located on a distant computer, you would first FTP to that site by typing *ftp madonna.coedu.usf.edu.* When asked for a login, type *anonymous*. You would then have access to all the files that are located in the directories designed for remote access.

Anonymous FTP commands can be used to transfer text files as well as program files. There is a multitude of programs ranging from HyperCard and LinkWay stacks to PageMaker files that can be easily transferred from one computer to another. In many cases, technology companies will put their updates, device drivers, or revisions on FTP for customers to retrieve. Because many of the files are quite large, they are usually compressed before they are transferred and must be de-compressed before they can be used.

Electronic file transfers are impacting both the development and delivery of training materials. Supporters suggest that companies will benefit from the Internet and decrease procurement cycles "up to 80 percent through online catalogs, ordering, and payment; ... shrink development cycles up to 50 percent; and accelerate time-to-market cycles through collaborative engineering and product implementation" (Internet News, 1994).

For example, resources exist on the Internet that will accept and print documents that are transmitted electronically. "This can dramatically reduce the turnaround time, and in fact can result in the delivery of the finished document the next day" (Maloff, 1994, p. 36). Similar services exist for the transmission of large files to optical disc reproduction services.

Complete training programs can also be transmitted via Internet or made available at remote sites. For example, *NetBiochem* contains learning materials for medical biochemistry and it is available through the Hahnemann University School of Medicine and the University of Utah School of Medicine. The current topics include Heme and Iron Metabolism, Macromolecules, and Nucleic Acids. The program offers graphics, sound, and limited interactivity. An entire course is planned for the future. (Address: http://www.hahnemann.edu)

NAVIGATING THE INTERNET

The multitude of information on the Internet can be overwhelming. In order to help people locate the desired information in a timely manner, several tools have been developed, including Gopher, Veronica, Archie, World Wide Web, and Mosiac.

Gopher. Gopher is a software navigation system of menu listings of the available files on the Internet.

Veronica. Veronica is a program that can search through the menus listed on the Gopher servers. After you type in a keyword, VERONICA will search the menu names and provide a list of the possible Gopher sites.

Archie. Archie is a program that searches the Anonymous FTP sites for a requested file.

World Wide Web (WWW). The World Wide Web links hypertext systems on the Internet. It contains documents that have links to other documents.

Mosiac. NCSA Mosiac is a unified interface to the formats used on the Internet. It provides powerful ways to search for, find, and share information in electronic documents (NCSA Education Group, 1993). Documents retrieved through Mosaic may contain text, graphics, audio, digital video movies, and hyperlinks to other documents.

CONCLUSION

The Internet is having a major impact on information search, storage, and retrieval. People throughout the world are discovering the cost savings, efficient response times, and ease of transmitting large, digital files on the Internet. The Internet is the first step in the information superhighway that is being built to connect the world instantly and internationally.

The Internet has its roots in education and research. Recently, the focus of the Internet has shifted to include commercial companies and services. There are an increasing number of tools on the Internet that can aid the design, development, and delivery of training in industry, military, and academic environments.

REFERENCES

Barron, A., Ivers, K., Hoffman, D., & Sherry, L. (1994). *Telecommunications: Ideas, activities, and resources.* Tampa, FL: Florida Center for Instructional Technology.

Calcari, S. (1994). Internauts: The next-generation. *Internet World*, 5(5), 80-83.

Dyrli, O. E. (1993). The Internet: Bringing global resources to the classroom. *Technology & Learning*, 14(2), 50-58.

Elmer-Dewitt, P. (1993). First nation in cyberspace. *Time*, 142(24), 62-64.

Internet News: CommerceNet Commences. (1994). *Internet World*, 5(5), 10-11.

Kent, P. (1994). *The complete idiot's guide to the Internet*. Indianapolis, IN: Alpha Books.

Maloff, J. (1994). The business value of internetworking. *Internet World*, 5(5), 34-39.

Mostafa, J., Newell, T., & Trenthem, R. (1994). *The easy Internet handbook.* Castle Rock, CO: Hi Willow Research and Publishing.

NCSA Education Group. (1993). *An incomplete guide to the Internet.* Champaign, IL: University of Illinois.

Thorell, L. (1994). Doing business on the Internet. *Internet World*, 5(5), 52-63.

EDUCATION, INSTRUCTION AND TRAINING METHODOLOGY SUBCOMMITTEE

Chair
Lou Hightower, Computer Sciences

Deputy Chair
Lundie Sherretz, Scientific Systems Company

Members
Bernie Edwards, USAF
Cliff Jackson, PULAU Electronics Corp.
Dr. Bill Melton, HQ TRADOC
Dr. Bruce Smith, Hughes Training Inc.
Dr. Emilio Mendoza, Gallactic Technologies
Dr. Jim McConville, Loral Defense System
Dr. Marcia Murawski, Vista Technology, Inc.
Dr. Robert Priseler, Star Mountain, Inc.
Dr. Steve Goldberg, Army Research Institute (STRICOM)
Eldon Riley, Comprehensive Technologies
Gareth Beitzel, USAF
Hal Hansen, US Army Training Support Center
J.C. Williams, Loral Advanced Distributed Simulation
Jim Jardon, Jardon & Howard Technologies
Jim White
Mike McGaugh, McDonnell Douglas Corp
Rick Beger, Naval Air Warfare Center Training Systems Division
Steve Husak, Dual, Inc.

Section 2
Table of Contents
Education, Instruction and Training Methodology Papers

INTERFACING INTERACTIVE ELECTRONIC TECHNICAL MANUALS WITH INTERACTIVE COURSEWARE

James D. Chenvert
Unisys Government System Group Training
Reston Virginia

ABSTRACT

The current trend of converting technical documentation to magnetic media could be a boon for training organizations, especially those developing Interactive Courseware. This paper describes the process of integrating Interactive Electronic Technical Manuals (IETMs) with Interactive Courseware (ICW), as performed during a U.S. Navy-sponsored demonstration project. The initial concept of putting technical references on-screen along with the interactive training material is introduced. The tools being used for the IETMs and ICW are listed with some supporting rationale. The roles of the ICW and IETM components are expanded upon to provide a background for the subsequent explanation of the implementation. The contrasting (or even conflicting) goals of ICW and IETMs are presented to illustrate the reasons for the implementation choices. The implementation itself is described with emphasis on the control of the IETM display from within the ICW. The problems associated with exercising that control are discussed and the solutions are presented. Finally, the resultant courseware is described. The description provides details of training screen layout including the instructions to the trainee, the navigation control bar, and the flow chart mechanization. The rationale for placement and sizing of the training window is also exposed. Additionally, the IETM window, with its size and placement are discussed.

OUTLINE: Interfacing Interactive Electronic Technical Manuals With Interactive Courseware

AUTHOR BIOGRAPHICAL DATA

James D Chenvert is a Principal Instructor. Jim began his career in training as a Navy Instructor at the Trident Training Facility at Bangor Washington. He later worked for Data Design Laboratories, the origin of OD 45519 – which was a precursor to MIL-STD 1379. Mr. Chenvert joined Unisys in 1982 and has been developing and delivering courseware for a variety of US and international customers ever since.

INTERFACING INTERACTIVE ELECTRONIC TECHNICAL MANUALS WITH INTERACTIVE COURSEWARE

James D. Chenvert
Unisys Government System Group Training
Reston Virginia

INTERFACING INTERACTIVE ELECTRONIC TECHNICAL MANUALS WITH INTERACTIVE COURSEWARE

THE CONCEPT

The concept was fairly straightforward. We would put the technical references needed by users of our interactive courseware right on the screen along with the technical training material. Our Interactive Courseware (ICW) is procedure-oriented maintenance training that relies heavily on the operational and maintenance documentation. We set out to demonstrate the capability of displaying the relevant portions of the technical manual, synchronized with the ICW lesson. We would somehow pass commands from the Interactive Courseware lesson to the Interactive Electronic Technical Manual instructing it to display in a window the portion of the technical manual being taught (see Figure 1).

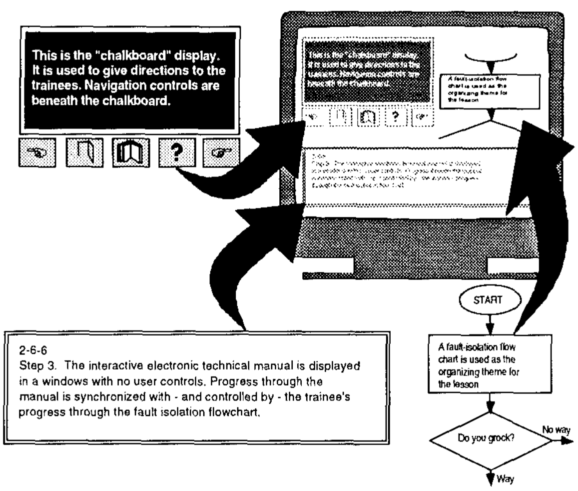


Figure 1. The concept for interfacing interactive electronic technical manuals with interactive courseware includes a technical manual window controlled by the trainee's actions in the courseware.

THE TOOLS

The Interactive Electronic Technical Manual (IETM) program we used for this project was a Windows-based program that enabled authors to develop, and users to access, SGML-compliant, interactive documents. Our Publications department had already chosen IDE/AS, which is a product developed by Unisys in Huntsville, Al, because it was easier and cheaper for authoring than some of the better known products. It was well-supported by Unisys and included the Dynamic Data Exchange (DDE) functions of Windows that made it possible to link with other programs.

The ICW authoring system we selected, after studying a wide field of contenders, was IconAuthor from AimTech. IconAuthor provides a rich, scriptless authoring environment with lots of power built in and extendibility for those situations where the built-in power doesn't meet some special need. We also found IconAuthor to be stable, handling a wide variety of platforms, and well-supported by AimTech.

Putting these two together in the Windows environment – which was pretty much made for this sort of thing – seemed to be an almost-trivial task. However, implementing a concept is almost never trivial and this was no exception. So much for the straightforward concepts.

THE ROLE OF THE ICW COMPONENT

The concept of interfacing IETMs with ICW has the two components fulfilling the roles they normally would except under slightly different conditions (see Figure 2). To expand on the concept just a bit, we know ICW takes advantage of computer technology to extend the range of the instructional institution. It allows an organization to project the learning experience into the field when required. Also, whether the training takes place within the walls of the institution or out in the field, ICW maximizes the influence of subject matter experts. That is, it gives a wider audience direct access to the subject matter expert's wealth of knowledge. Depending on the authoring system and the approach taken, ICW may also extend the communication capabilities of the courseware designer. The multiple media available through an authoring system like IconAuthor allows the designer to tune the delivery to the unique requirements of the lesson. Because of these characteristics, ICW had already been chosen as one of the media to be used to meet the customer's training requirements.

THE ROLE OF THE IETM COMPONENT

Interactive Electronic Technical Manuals are designed to extend the reach of the user. They allow the user to rapidly and easily gain access to relevant information. Technical information, drawings, procedures, and the like become much more useful tools. The user can put the documentation to work without being concerned about how many volumes need to be transported and whether or not the right volumes have been chosen. While electronic technical documentation can make storage and retrieval significantly easier, the manner in which the documents are captured and stored can make all the difference in the world. If the material in the IETM is scanned as a graphic, it cannot be searched by content. It can only be accessed by some external index system – page, paragraph, or whatever the scanned unit was – as opposed to being word-searchable.

With a document that is word-searchable, the user is able to search for a word like 'widget' and get a list of all occurrences of the search word. The Structured, General Mark-up Language (SGML) standard allows for text to be made 'hot.' Hot text is indicated by a distinct color and/or font style and is linked to material that expands on the indicated word or phrase. If the user clicks on the hot text 'widget,' the document reader program goes to the portion of the document where 'widget' is defined. Likewise, if the text of a paragraph refers to a table or figure, the user can view that table or figure by clicking on the reference to it. This is a great usability stride that just isn't available if a document is not word-searchable. The minimum standard for electronic documents should therefore allow for retrieving any numbered element, meaning chapter, section, paragraph, table, figure (and sheet), and step. Since a word-searchable IETM implementation had already been chosen for our customer, the prerequisites for our demonstration were already met.

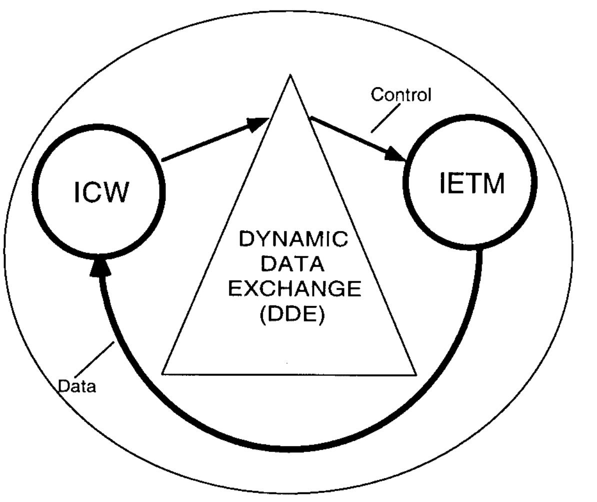


Figure 2. The original concept was to have the ICW sending control (positioning) data to the IETM via Dynamic Data Exchange and the IETM responding with technical information for use by the ICW.

THE IMPLEMENTATION

The marriage of the two technologies seemed to be a match made in heaven. Oddly enough, though, it wasn't. The goal of the IETM is to provide users with all the information they want, on demand. IETMs are quite successful in meeting that goal, too. However, technical manual excerpts in ICW are very specific and must relate to the topic at hand to prevent a lesson from becoming a free-form exploration of whatever tickles the user's fancy. There is nothing wrong with free-form exploration, its just that it isn't germane to a procedurally-oriented ICW lesson. The structure of the ICW must keep track of a student's progress through a lesson and make whatever corrections are required. That becomes more and more difficult as the student gets further away from the directly-related reference. Knowing which portions of the technical manual were referenced might provide insight to the user's thought patterns and thereby facilitate a critical performance analysis. However, for the purpose of providing feedback to users to constrain them to the correct fault isolation path, it was sufficient to know that the users were not following the procedural path. With a courseware-driven IETM, trainee progress information was available.

That is why the characteristics of general-use IETMs are in conflict with the characteristics of IETMs that would be built solely for use in conjunction with ICW. The delivery package for the general-use IETM strives to make any and all information available while the delivery package for use with ICW must constrain the user to those passages directly related to the problem at hand.

In addition, the controlling entity is different for general-use and ICW-related types of delivery. For general-use IETMs, the user controls what is displayed and when it is displayed. For ICW-related IETMs, it is the courseware structure that must control the IETM presentation. Or, at least, that was the design choice we had made (see Figure 3). It could certainly be argued that the ICW user should also control the IETM. We felt, though, that the manipulations required for the IETM, when added to that required for the ICW, would start to distract from the fault isolation lesson being presented.

While IETM programs could probably be adapted to report the user's selections, they don't at this time. IETMs do track a user's selections to facilitate backtracking but there isn't a built-in way to export that information (at least, we didn't find any). The fact we couldn't determine a user's path from outside the IETM was the reason we didn't attempt to use Object Linking and Embedding (OLE) instead of DDE. As it turns out, the OLE route wasn't open to us anyway since neither the authoring language nor the IETM supported OLE.

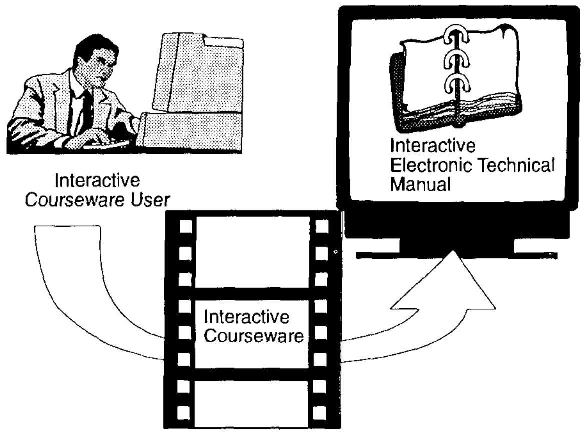


Figure 3. The user controls the presentation of the courseware. The courseware, in turn, controls the display of data from the interactive electronic technical manual.

Controlling IETM Presentation

Our problem, then, was to be able to control the presentation of IETM data from within the structure of an ICW lesson. We also needed to control the attributes of the IETM reader window so that it worked harmoniously with the ICW lesson. The problem of controlling the IETM presentation proved to be more difficult than we expected. We had decided early-on that we would use the Windows Dynamic Data Exchange (DDE) facility to implement the control of the IETM. The DDE facility allows the passing of data between programs. Since the ICW lesson we were linking was tied very closely, even lock-step, with the procedure, the correct technical manual reference at any point in the lesson was easily obtained. All we needed to do was pass the correct reference to the IETM program and display the relevant passage in a window on the screen.

The Problem With Implementing the IETM Control. The good news was that the developers of the IDE/AS program included a DDE function, even though no users had yet requested that functionality. The developers had experimented with DDEs and incorporated some basic functions in anticipation of their user's needs. The bad news was that the DDE functions did not support document positioning from an external program (in particular, ICW). That meant we could pass messages between the ICW and the IETM, but we couldn't use those messages to position the document where we wanted it. We found, though, that the IETM program could be positioned by sending the rudimentary movement commands, right and left arrows, through a DDE function called SendKeys. The IconAuthor program supported several DDE functions but, unfortunately, SendKeys wasn't one of them.

Solution to the IETM Control Problem. The solution to the problem of controlling the presentation of IETM data was to use an intervening program to receive the desired document position from the ICW (developed in IconAuthor) and translate that position into relative movement commands to be sent to the IETM via SendKeys. Microsoft Word has a powerful macro language based on the Visual Basic programming language that can be called up by another program through DDE functions. IconAuthor supported the DDE functions necessary to start Word macros. The Word macro language supported the SendKeys DDE function to which the IDE/AS program responded. So, to move the document from paragraph 1-1-4 to paragraph 1-1-5, we would send a command from IconAuthor to Word telling Word to run a macro called 'Right' one time. The Right macro would use the SendKeys DDE function to send the code for the right arrow to the IDE/AS program which would respond as if a user had pressed the right arrow. That is, it would move from paragraph 1-1-4 to paragraph 1-1-5.

The Expeditious Solution. We employed Word as an intermediary as an expeditious solution to the immediate problem of controlling access to IETM data from the ICW. While this scheme was functional, it clearly wasn't a long-term solution. You can imagine the awkwardness of this solution when it was necessary to go from paragraph 1-1-4 to paragraph 6-1-4. Getting there one key-stroke at a time is cumbersome even for a computer. The scheme also breaks down when the reference is not contained in the same file as the current paragraph. The IDE/AS program allowed new files to be loaded through the DDE functions it supported, but the logic necessary to resolve all the possible document shifts and right or left movements was more complex than could be rationalized for a demonstration. Especially when we felt the real solution was quite different from this scheme. On top of all that was the added burden on computer resources of keeping Word running in the background while IDE/AS and IconAuthor ran in windows. It was a good thing the demonstration would be short.

Other IETM Display Wrinkles

The very fact we were using a Windows-based reader program for the IETM was a source of difficulty in designing the user interface. The normal Windows program window has facilities for closing and resizing the window. A user who did that would have access to the hidden workings behind the ICW user interface (sort of like Dorothy throwing back the curtain of the Wizard's control booth, don't you think?). We avoided this undesirable situation by cheating a little. Our interface design had the upper half of the screen occupied by the 'training' display and the lower half occupied by the IETM display. We had excellent control of the training portion of the display but only manual control of the IETM portion. We made sure the training display overlapped the IETM display by an amount that made the window controls inaccessible to the user. We also sized the training display so its window controls were above the top of the visible display. Using this technique, we were able to eliminate the potential for users to be led astray by the Windows capability to alter program appearance.

Controlling IETM Display Attributes

The problem of controlling the attributes of the IETM window centered around our belief that the ICW user shouldn't directly control the IETM data display. We wanted to have the user interface only with the ICW window and we would have the ICW drive the technical manual to the proper position.

The Problem With Controlling the IETM Display. The difficulty we encountered was the reader program wasn't designed for a 'remote control' application. This meant there was no facility in the program to control such things as window size and position, and whether or not the window could be re-sized or minimized by the user. Worse, the availability of the controlling buttons and menus was determined by 'tags' embedded in the text, along with tags that controlled the appearance of the text. That meant we couldn't send a command to the reader program to make the menus and buttons unavailable. The user could therefore disrupt the lesson flow simply by using some of the features of the reader program. It also meant that, since the availability of hot text was controlled by embedded tags, we couldn't make the hot text inactive. The hot text would allow the user to jump around within the manual just as the menus and buttons would.

Another Expeditious Solution. We asked the IDE/AS developer if the program could be modified to provide control of the IETM window and the text attributes from the program instead of through the embedded tags. The developer assured us it could be done, so we accepted that at face value and, for the purposes of the demonstration, modified the embedded tags to emulate a modified reader program. We copied all the required technical manual data to a separate file and turned off the button bars and menus and made the hot text look like all the other text. Obviously, we wouldn't do this for a deliverable piece of courseware but for the purposes of evaluating the viability of this concept, we accepted this as expeditious. The reality didn't exactly match the concept (see Figure 4).

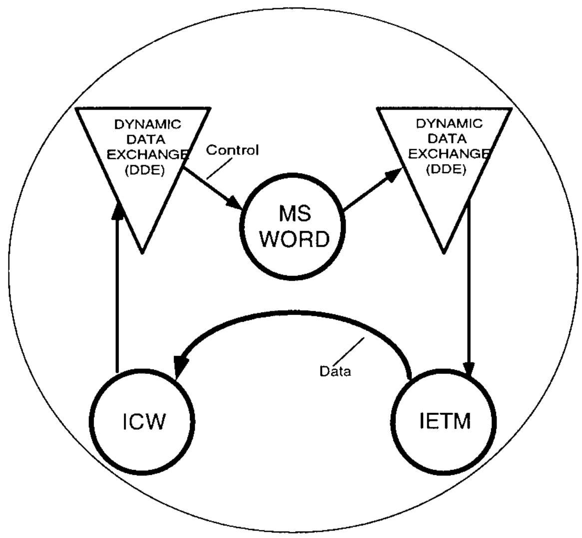


Figure 4. The reality of making the ICW and IETM interact included the use of an intermediary (Microsoft Word) as a translator for positioning commands from the ICW to the IETM.

Unrestricted User Access to Manuals

It should be noted that the lengths we went to in removing user control of the IETM was only for maintaining the integrity of the training delivery. We also provided a button which gave the user complete access to the fully-functional technical manual. When the user selected that button, another instance of the reader program was invoked and the lesson was suspended. When finished with the technical manual, the user exited and was returned to the lesson at the point where it was suspended (as opposed to where the user was in the technical manual).

THE RESULTING DEMONSTRATION COURSEWARE

The courseware we put together to demonstrate the interaction between ICW and IETMs was a fault isolation exercise for the second-level (depot) maintenance technician. We used a problem recently encountered in the maintenance shop. The fault isolation procedures included an overall fault isolation flowchart which referred to individual procedures in the manual in addition to common procedures in another manual. To implement the ICW/IETM approach, we chose to mechanize the fault isolation flowchart. That meant producing the graphics to represent the flowchart, one flowchart symbol per step. The graphics were fairly straight-forward and organizing the lesson along the lines of the flowchart made sense because we were going to be implementing the logical choices made on the flowchart anyway.

The 'training' portion of the display comprised a 'chalkboard' and the flowchart symbols along with a navigation control bar. The chalkboard resembled a real chalkboard and carried directions, messages, and status information to the student. The flowchart symbols, as previously discussed, were mechanized versions of the fault isolation flowchart. The navigation control bar gave the user control over direction and speed of the lesson and access to facilities like 'help' and the fault isolation scenario (see Figure 5).

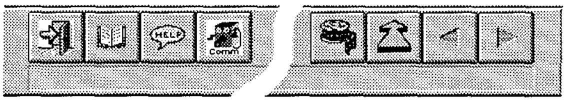


Figure 5. The navigation bar, though it looks harmless enough, triggers a series of events that synchronizes the IETM presentation to the ICW position each time the user makes a selection.

The ICW user was presented with a scenario at the beginning of the lesson which provided the impetus for performing fault isolation as well as the initial conditions of the affected equipment. Acting on the initial indications, the user proceeded (in accordance with the documented process) to the fault isolation flowchart. The fault isolation flowchart asks questions to determine the most likely problem and directs specific procedures be performed, such as turn-on and turn-off, resistance checks, cable inspections, and so on. Each of the directed actions had a corresponding technical manual procedure. When the ICW user selected an action, the IETM was driven by the courseware to the appropriate paragraph, step (the procedures were documented in paragraphs), table, or figure number. Comprehension checks were instituted to ensure the user understood the lesson. When a user made an incorrect choice on the comprehension check or in choosing the next fault isolation step, immediate feedback was given, the location of the correct information was displayed, and the user was brought back to the step where the correct information was in the on-line technical manual.

While a technician using paper technical manuals would have had at least two volumes open and would be flipping back-and-forth between them, the Interactive Electronic Technical Manual tracked through the fault isolation process effortlessly.

SUMMARY

The combination of Interactive Courseware (ICW) and Interactive Electronic Technical Manuals (IETMs) in technical training is a natural evolutionary step in the procession towards a paperless operating environment. The two technologies weren't developed with the idea they would become a symbiotic pair, but you have to wonder why not. The goal of the demonstration was to show the viability of using Interactive Electronic Technical Manuals in conjunction with Interactive Courseware. The results showed the concept to be valid – but the implementation suffered from software that was not ideally suited to the task. A relatively small effort to tune a reader program to the needs of ICW would go a long way. The software issue, though, is a small stumbling block. The concept is viable as we demonstrated – and inevitable.

ADVENTURE GAMES FOR TECHNICAL EDUCATION

Henry M. Halff
Mei Technology Corporation
San Antonio, TX

Abstract

This paper describes the use of adventure games for technical and scientific education. The topics most appropriate for instruction via adventure games are those such as chemistry and physics that require knowledge of abstract concepts and mastery of advanced problem-solving skills. Adventure games that teach such topics can be constructed as a network of rooms in which each room represents a concept or skill and the paths among the rooms reflect the conceptual structure of the subject matter. Each room offers the player an opportunity to practice the focus skill or explore the focus concept for the room. Ancillary support for learning can be provided via conventional computer- or text-based instruction, hypertext, and visualization techniques.

Games of this sort offer signal advantages over conventional computer-based or classroom instruction. Their motivational advantages are clear. Properly constructed they allow the student to conceptualize the structure of the subject matter in terms of the game topology, thus bringing the power of spatial cognition to bear on the difficult task of conceptual organization. The adventure environment can immerse the student in the subject matter in a way that is often impossible in the real world. Instructional exercises can be focused on critical learning objectives thus increasing time on task. Instruction can be adaptive so that students devote only the time needed to master the subject matter. Visualization techniques can be used to convey difficult abstract concepts.

Cost effective development of computer games can only be accomplished if the dual nature (instruction and entertainment) is recognized. The market for instructional adventure games is often not the same as the market for commercial games. Special mechanisms (e.g., hypertext) are required to meet instructional objectives. Prototypes and other mechanisms needed to ensure that instructional methods and content are effective.

Biography

Dr. Halff is a research psychologist with twenty-five years of experience in learning, instruction, and instructional technology. His broad range of skills covers high-level research planning, bench-level R&D, and instructional design. In his current position at the Mei Technology Corporation, he manages a research program on generative, knowledge-based instructional technology. He also conducts research on the use of computer games for science education.

Before joining Mei Technology in 1993, he owned and operated a successful consulting firm, Halff Resources Inc., where he designed and developed both conventional and computer-based instruction for maintenance, management, sales, and other areas. Prior to founding Halff Resources in 1984, he worked for seven years as a scientific officer in the Psychological Sciences Division of the Office of Naval Research. There, he developed a reputation in scientific, military, and government communities for his management of research programs in the rapidly advancing fields of educational technology and cognitive science. These highly acclaimed programs reflect his expertise in computer-based instruction, applications of artificial intelligence to instruction, and applied aspects of cognitive science. He came to the Office of Naval Research from the University of Illinois at Urbana-Champaign, where he was an Assistant Professor of Psychology (1970–1976). His activities there covered mathematical and cognitive psychology, and he specialized in models of learning, decision theory, computer-based research, and statistics. He was an NIMH postdoctoral fellow at the University of Michigan's Human Performance Center in 1969–1970. In 1969 he earned a doctorate in Psychology from the University of Texas at Austin.

ADVENTURE GAMES FOR TECHNICAL EDUCATION

Henry M. Halff
Mei Technology Corporation
San Antonio, TX

INTRODUCTION

In the Summer of 1991 a colleague who works for the Navy called me to ask if I would be interested in developing a computer game to teach budding avionics technicians about basic electricity and electronics. I asked, "You want to pay me to do this?"

"Sure," he replied, "I figure It will take you and a couple of graduate students." In the Summer of 1993, a team considerably larger than me and "a couple of graduate students" delivered a version of a game known as *Electro Adventure* that taught part of what students in the Navy's Avionics (AV) "A" School needed to know about capacitors and capacitance.

This paper is partly the story of the development of *Electro Adventure* and partly an exposition of the lessons I learned in the course of that development. You should know that, by choice, I functioned as *Electro Adventure*'s instructional designer. Thus, the treatment below focuses on instructional design. Our software designer and game designer would give entirely different stories. The Navy Personnel R&D Center conducted an empirical investigation of the game's effectiveness and will have yet another story to tell.

I hope to package the following remarks in a form that will be useful to those developing similar products, namely, computer-based adventure games for technical and scientific education. To determine whether you are among this audience, you will need to know what I mean by "technical and scientific education" and by "computer-based adventure game."

The Subject Matter

Technical and scientific topics of the sort addressed by *Electro Adventure* have several interesting characteristics.

Technical and scientific topics rank high on the difficulty scale. Parents of high-school students, I suspect, hear more complaints about Chemistry

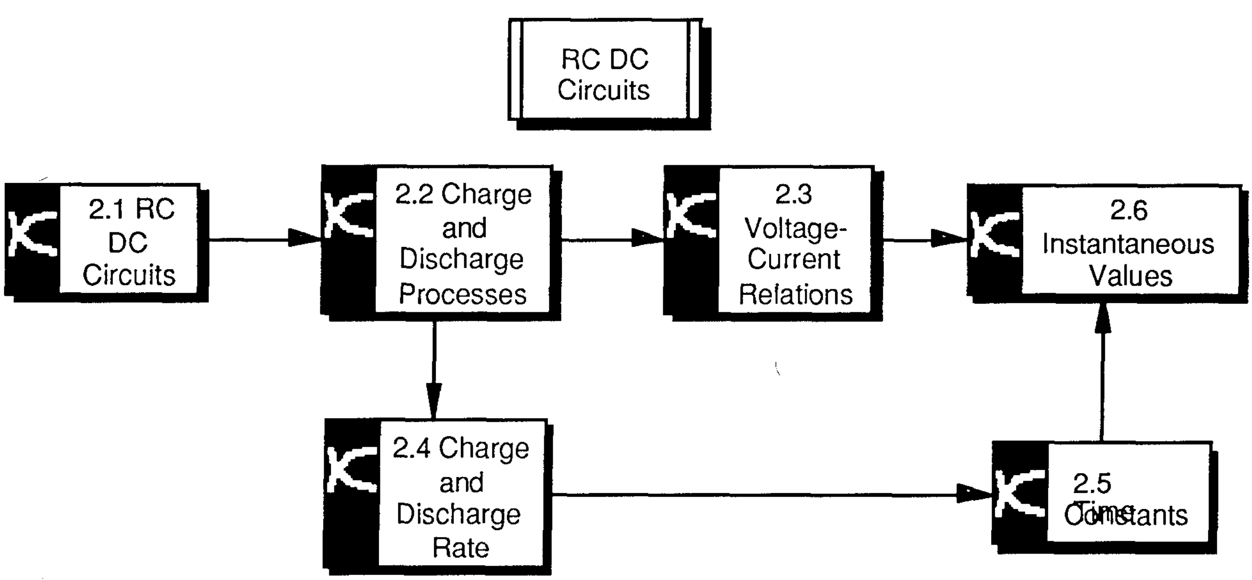


Figure 1. Concept Network for a Segment of *Electro Adventure*.

than Social Studies, and qualifying for technical schools in the services is a considerable challenge. Hence, any instructional approaches that bring these subjects within the reach of more students should have a considerable market.

Of more instructional relevance is that these topics can often be characterized by a network of concepts. Figure 1 shows the network that makes up some of the concepts addressed in *Electro Adventure*. Note that a prerequisite relation structures the network in that some concepts cannot be mastered before others. For example, one cannot understand how RC circuits charge and discharge (2.2 in Figure 1) before one knows what an RC circuit is (2.1 in Figure 1).

Another important characteristic of technical and scientific education is an emphasis on quantitative problem solving skills. Even the most elementary chemistry course requires students to solve problems involving combining ratios of elements and concentrations of chemicals in solution. Students in the Navy's AV "A" course must be able to find the equivalent capacitance of multi-capacitor circuits and to determine instantaneous circuit quantities at particular points in charging and discharging cycles of resistor-capacitor (RC) circuits.

A fourth important characteristic of the subject matter found in technical and scientific education is the importance of qualitative knowledge. Much of the difficulty in mastering technical subjects lies in the importance of abstract variables such as electrical current, and the constraints among them. A primary goal of technical education is that of teaching students to reason about the behavior of these variables, and the difficulty in acquiring these reasoning skills has much to do with the fact the variables are several steps removed from actual experience.

It was these four characteristics of technical education—their challenging nature, the structured character of the concepts, the critical role of quantitative problem solving, and the importance of qualitative reasoning—that struck me, and still strike me, as making this type of subject matter promising for the application of computer-based adventure games.

Computer-Based Adventure Games

Computer-based adventure games are computer games in which the player assumes the role of a character in some fictional scenario. This character can move around in the environment defined by the scenario, carry and manipulate objects in the scenario, and converse with other characters. The environment itself consists of a network of regions, typically called "rooms." In Electro Adventure, the environment was a ship named the Electro. Figure 1 shows the topology of one section of the Electro.

Some games such as Super Mario Brothers rely heavily on psychomotor skills; others rely on the player's ingenuity and problem-solving skills. It is this latter sort of game that is of interest here since they offer the most promising opportunities for education. Examples include the King's Quest series published by Sierra On-Line and Loom, published by LucasFilm. These games have many goals including survival, material gain, amassing points for accomplishments, exploration of the environment, and achieving some particular objective such as rescuing a damsel in distress. Strong motivational characteristics constitute one of the many reasons for applying adventure games to technical and scientific education.

To elaborate, two well known limitations of conventional training and education are lack of opportunities to practice the skills being taught and failure to sustain motivation over the long periods needed to achieve competence in the target skills. Adventure games address both of these problems by providing for extensive practice in an environment that is almost certainly more motivating than conventional education and training. It is also possible in these games to present to students situations and problems that are not feasible to present in the "real world."

In addition, all of the benefits available in other multimedia instructional software are also available in adventure games. Chief among these benefits are opportunities for the use of hypermedia and the application of animation to visualization of complex scientific concepts.

The fantasy aspects of adventure games also offer unique advantages for instruction. Students are able to practice skills and apply knowledge in semi-realistic environments. Although these environments do not offer all of the features of the real world, they can be constructed to convey important lessons about the application of technical skills in real world situations.

In some respects, fantasy environments are superior to real-world environments in that they can be designed for an optimal instructional sequence and to maximize the student's time on critical learning tasks. In fact, I feel that the best of these games provide a level of immersion in the subject matter that is impossible to achieve in the real world, classroom, or laboratory.

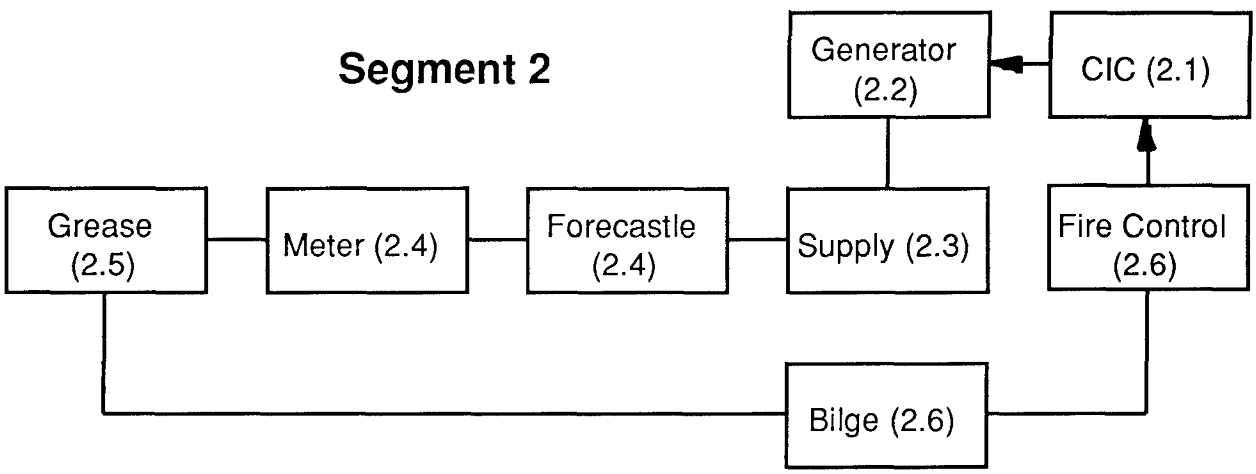


Figure 2. Topology of the Section of the Electro Addressing Topics Shown in Figure 1.

Finally, adventure games offer a natural way of mapping the structure of the subject matter to the topology of an adventure game. In *Electro Adventure*, for example, the tasks or challenges found in each room address a single topic in the curriculum. Furthermore, the topology of the game is derived from the network structure of the topics. Figure 2, for example, shows the topology of the section of the Electro that addresses the topics on RC DC Circuits shown in Figure 1. Notice that students cannot enter the rooms addressing higher level topics before they have visited the rooms addressing their lower-level prerequisites. Mapping game topology to the structure of the material in this fashion is a natural way of enforcing prerequisite relations. Perhaps more importantly it allows students to bring spatial cognition to bear on the difficult problem of developing a conceptual structure for the domain under study.

ADVENTURE GAMES FOR INSTRUCTION

Adventure games designed for instructional purposes will, of course, differ in many ways from those designed for entertainment, and the topology-topic correspondence is only one distinguishing characteristic of the former. A picture of these distinguishing characteristics emerges from an examination of *Electro Adventure.*

The Scenario

Electro Adventure's scenario centers on a ship, the Electro, that, through a mishap has been transported from the future to the present time. The mission of a player is to find the resources needed to repair the ship's computers and return her to the future. Accomplishing this mission requires the traversal of a number of rooms, or rather compartments, on the Electro. Each of these rooms addresses a single concept, topic or instructional objective, which we will call the room's *focus.*

Instructional Mechanisms

Figure 3 shows the student's view of a typical room in *Electro Adventure.* The student can explore the room, discover objects therein, containers that themselves contain objects, and characters of various sorts. Instruction in the room's focus is delivered in two ways.

- The student may be required, on his or her own, to discover certain technical tricks needed to survive in or escape from the room. Failure to use proper safety procedures may, for example, result in the student's death. As another example, the student may be required to correctly combine materials and parts found in the room in order to manufacture some equipment needed in the game.

- In most rooms, the student is also required to solve a series of technical problems of the sort typically found in textbooks. Figure 4 illustrates such an exercise from *Electro Adventure.* These exercises have several important characteristics.

  - Students are provided with a special interface to the problem-solving environment by zooming to a *problem view* like that shown in Figure 4.

  - The sequence of problems provided to the student can be graded in complexity.

  - The parameters of each problem are randomly chosen so that students can obtain additional practice by replaying the room.

    An additional benefit of this feature applies when the game is played in a social setting. Students helping each other through the game cannot provide answers to individual problems since no two students see the same problem. Hence, help and advice must focus on the strategies and procedures for solving the problems.

  - A performance criterion can be imposed to ensure mastery of the skills addressed by the exercise.

Instructional support for each room's focus is also provided by a conventional *Computer-Based Training (CBT)Lesson* for the room. The lesson consists of a series of frames containing instruction on the room's focus. Certain interactive devices such as multiple-choice questions are used to maintain attention and ensure comprehension of the material.

A *Hypermedia System* is also available to the

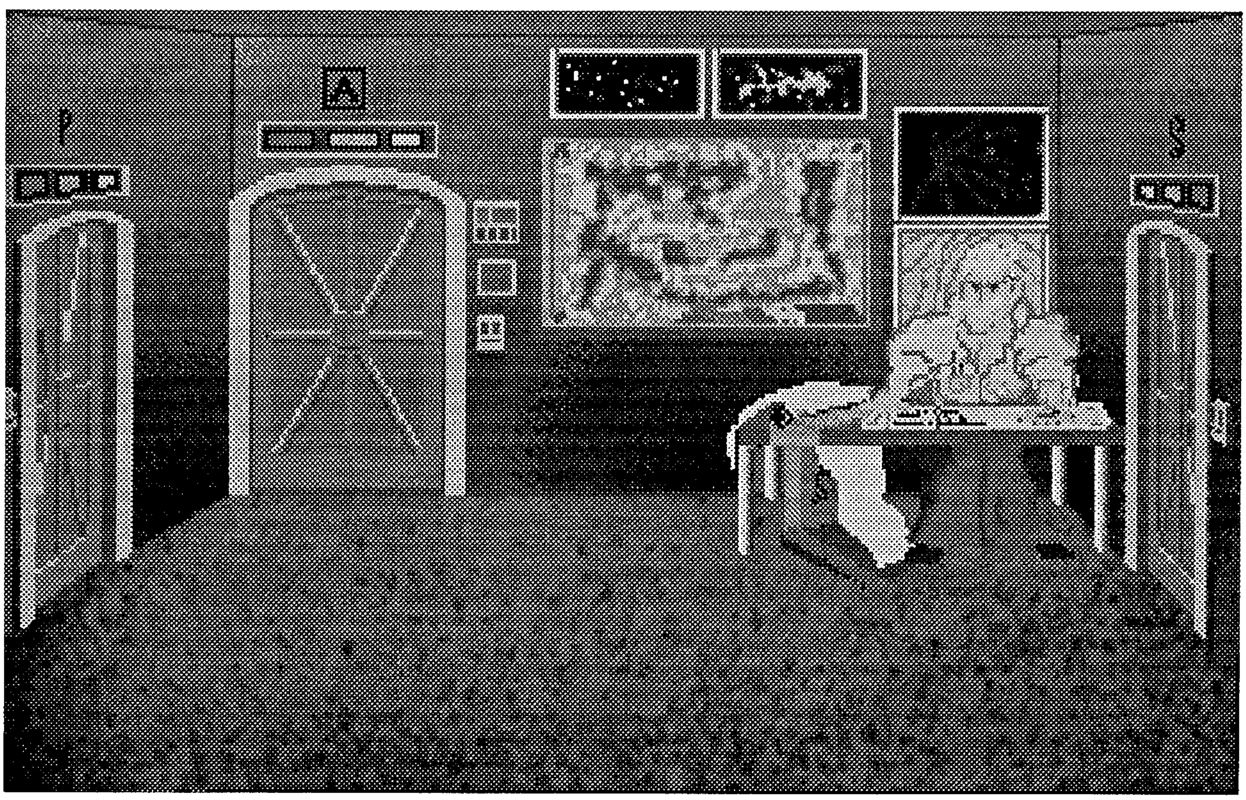


Figure 3. Typical Room from *Electro Adventure.*

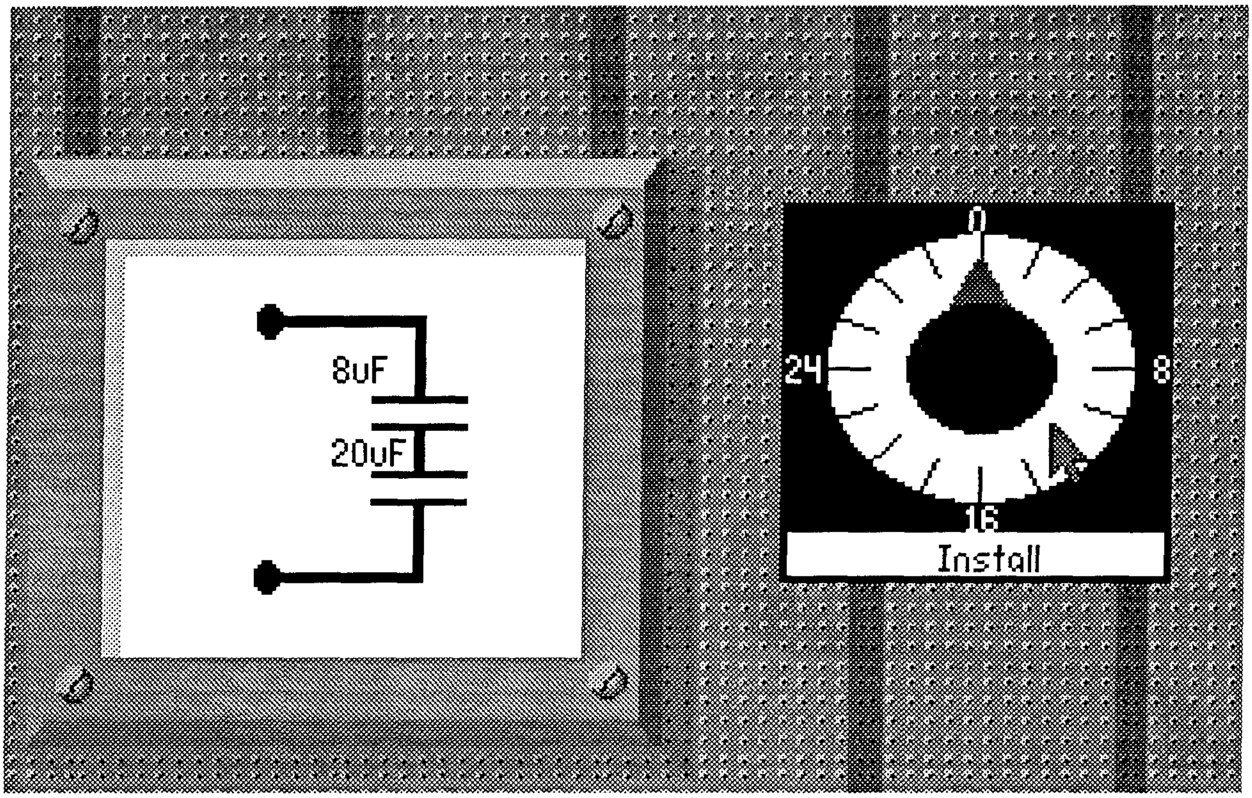


Figure 4. Typical Problem View from *Electro Adventure.* (Note: In this exercise the dial must be set to the equivalent capacitance of the circuit in the panel.)

students for purposes of remediation, review, and exploration of the subject matter. The hypermedia system chosen for *Electro Adventure* is an adaptation of a product called *Thoughtsticker* developed by Paul Pangaro. The system consists of a set of *topics* defining the domain and a network of *presentations* used to present information on these topics to the student. The mapping of topics to presentations is many-many and complex, but the complexity is typically hidden from the student. Students using the hypermedia system are given a menu of topics relevant to the focus of the current room. The system, based on its past interaction with the student, uses heuristics to choose the most appropriate presentation for any chosen topic. Embedded in each presentation are links to other topics, and hence other presentations.

Employed by both the CBT and hypermedia system are *Visualization* techniques used to illustrate complex technical concepts. In *Electro Adventure*, for example, color and animation are used to show how current and voltage change over time in various parts of an electrical circuit. The general principles of computer-based visualization are the use of color and animation as concrete representations of abstract quantities, and the capability of the student to control the parameters of the system being visualized.

Finally, a student interface is required along with other mechanisms to support the operation of the program itself. Among the components of this *Operating System* are a tutorial or guide to the student interface, mechanisms for saving the game state and other data, and a means for setting game preferences such as sound level.

THE DEVELOPMENT PROCESS

Design and development of a product such as *Electro Adventure* can be a major effort. In this respect, adventure games are more like movies than computer programs. The credits that roll at the end of a game such as Kings Quest IV list a host of artists, musicians, animators, and a few programmers. Major software houses such as LucasFilm and Sierra On-Line have developed sophisticated facilities for production of remarkably lifelike scenery and action.

Fortunately, non-commercial instructional games do not necessarily require the extensive effort and facilities involved in commercial efforts. The latter must compete in the marketplace where an extra 10% in sophistication and realism can mean the difference between success and failure. Instructional products often have no competition apart from conventional classroom or computer-based training. The degree of realism and sophistication need not be as great as that of commercial products.

Talent and Resources

Still, development of any sort of game is beyond the capabilities of one person or even an instructional designer working with a couple of graduate students. The development team for *Electro Adventure* gives a good indication of the kind of talent needed to develop an instructional adventure game.

- A sponsor, the U. S. Navy[1], provided funding. In addition it defined the overall research goals of the effort.
- A client, the Chief of Naval Technical Training provided the particular instructional objectives to be addressed in the game. In addition it provided subject matter expertise and an environment for evaluating the product.
- The program was managed by the University of Central Florida's Institute for Simulation and Training (IST), the Navy's prime contractor. Dr. Bruce MacDonald was the project manager for IST.
- An instructional designer (myself) worked with an instructional developer, Ms. Catriona Borbato, at IST to provide the instructional design.
- A game designer, Mr. Howard Delman, developed the fantasy used in the game and provided guidance on software development.
- A software design and development team at IST, headed by Mr. Ron Klasky, developed the software for the game.
- Computer-graphic artists (under the direction of Dr. Jackie Morie at IST) and a musician provided art for the product.
- A hypermedia designer, Dr. Paul Pangaro, helped with the implementation of the hypermedia system.

The Process

The process used to develop *Electro Adventure* was chaotic, to say the least, and only informative as a guide to what not to do. I therefore take the liberty of describing not what how the game was developed, but rather, how I would develop it if I had the opportunity to start over.

Development Facility. Create a development facility. This facility should have the following elements.

- An *Adventure Language* should be adopted for describing adventure-game scenarios. At least one such language (TADS) already exists[2]. Although it is specialized for text adventures, it is extensible, and, with some modification, it should serve to describe the adventure games of the type considered here. The language not only provides an important prototyping tool but also formally defines the possibilities for the game scenario.
- A *Student Interface* should be designed that implements the semantics of the adventure language. That is, the

---

[1]Three Navy organizations contributed to this effort under the umbrella of a program known as the Skill Enhancement Program. Participating were the Navy Personnel R&D Center, Op-112, and the Chief of Naval Education and Training. The Naval Training Systems Center managed the effort for the Navy.

[2]TADS: The Text Adventure Development System—Version 2.0, a shareware product (© 1992) by Michael J. Roberts.

specification should denote how actions in the game are depicted on the display and how actions by the student influence the course of the game.

- A *Hypermedia System* should be provided. This component may, and probably should, be nothing more than a commercially available hypertext shell.

- A *CBT System* should be adapted to provide the lessons addressing the focus of each room. As with the hypermedia system, this component could be an available commercial product.

- A complete description of the student interface and support system should be provided in the form of a *Users Reference Manual* This manual should describe all general options available to the student at any time and the effects of exercising those options.

Instructional Design. The instructional design determines the content of the game. It instructional design should have the following components.

- A *Conceptual Map* details the instructional objectives of the game. It should consist of a network like that of Figure 1 together with a narrative description of each concept in the network. Also required for accountability purposes is the relationship of each concept to the course's instructional objectives. Typically, concepts will be cast at a greater level of detail than instructional objectives so that several concepts will be required to cover each instructional objective.

  The conceptual map is the basis for all subsequent development. It's structure determines the topology of the game scenario. At a more detailed level of elaboration, it determines the structure of the hypermedia system. Since the CBT component is keyed to rooms, the conceptual map also determines its structure.

- The *Adventure Scenario* describes the adventure's environment, objects, characters, and other characteristics. It consists of four components:

  - a map showing the game's topology (see Figure 2),

  - a narrative description of the game's rooms, characters, objects, and special effects,

  - a formal description of the game in the Adventure Language described above, and

  - sketches illustrating the rooms, objects, characters, and special effects.

- The *Exercises Design* describes the exercises to be delivered in each room. It includes

  - narrative descriptions of the exercises,

  - the procedures to be used to generate exercises on-line,

  - illustrations of the student interfaces to the exercises, and

  - prototypes of the exercise-generation procedures. Such prototypes, although not strictly necessary are the most efficient way to refine the exercise-generation procedures and to present them to subject-matter experts for review.

- The *CBT Lesson Scripts* are a complete set of scripts for all of the CBT Lessons, including the text of each lesson and sketches illustrating art and animation to be employed in these lessons.

- The *Hypermedia Design* consists of a list of topics to be used in the design, a description of the collection of presentations, and a specification of the network defining the structure of the hypermedia document.

Development. If the development facility is properly designed, the development process itself should be almost routine. Major steps in development include

- the conversion of room, character, and object sketches to computer graphics for integration into the game,
- the development of software for the exercises based on the exercise design,
- the development of CBT lessons based on their scripts, and
- the entry of data into the hypermedia system.

LESSONS LEARNED

The product and process described above illustrate one approach to the design and development of instructional adventure games, and, in fact, the approach is by-and-large untried. In order to provide a larger perspective, it is fitting to conclude with some of the lessons that we learned in the course of developing *Electro Adventure*.

- Pitch the product design to the intended market. The initial design goal for *Electro Adventure* was a product that would be competitive with the best found in the commercial market. As the result, many expensive artistic effects were included that contributed little to effectiveness. A more appropriate design philosophy would have recognized that the real competition for *Electro Adventure* is an instructor and chalkboard.
- Constrain the complexity of the game scenario. The game scenario in *Electro Adventure* was written without any regard to the programming challenges that it might present. Indeed, because of lack of experience, the programming team could not even anticipate which aspects of the design would prove to be difficult. By using a formal language, you can impose a discipline on the game design by restricting the scenario to one that can be represented in the Adventure Language and its student interface. You can also simplify software development by explicitly separating room exercises from the adventure's scenario.
- Provide tools and products for early review of the design. Most of the paper design documents proposed here were also produced in the course of designing *Electro Adventure*. However, these paper descriptions were completely opaque to the Navy's subject matter experts (and to many on the design team itself). As the result, no meaningful reviews could be conducted until the project had been committed to code, at which point changes were exceedingly expensive. To remedy this problem, you can to provide more sketches and working prototypes of the exercises so that quality can be assured before programming begins.
- Include all components in the design. In *Electro Adventure*, as in other projects, major features, most notably the hypermedia system, were inserted into the project after the initial design and some development were completed. The inclusion of these unplanned for capabilities were the source of considerable stress on the product development process. There are considerable benefits to committing the project to the design as written.

As a final note, you may be interested in a preliminary look at the evaluation conducted by the Navy as mentioned above. This evaluation consisted of a single experiment in an AV "A" school. In this experiment, Electro Adventure was pitted against stand-up instruction and two other Computer-Based Instructional (CBI) systems developed at NPRDC. The results were mixed.

One indication, of singular importance to the students, is performance on the 40-item test that they are required to pass. Table 1 shows the mean percent correct on this test. for each group. *Electro Adventure* did not fare too badly by this standard. Worth noting in this regard is that we had access to the testing instrument during development and were amply warned of the dire consequences should students fail the test.

This was not the case with another test developed at NPRDC by the same team that developed both of the CBI lessons. Results on this 47-item instrument show some superiority for both of these CBT conditions over stand-up instruction and the game. The test was developed in the final stages of instructional development

Table 1
Percent Correct on Evaluation Tests

Group	Stand-up	Electro	CBI 1	CBI 2	Std Dev
Unit Test	89	86	84	81	11
NPRDC Test	68	67	75	74	13
IMMS	12	13	12	14	na
N	13	23	24	20	

and therefore had no influence on the development of *Electro Adventure.*

NPRDC also administered a motivational measurement questionnaire, the *Instructional Materials Motivational Scale* (IMMS). As Table 1 shows, Electro Adventure did not show any particular superiority on this instrument. To understand this result, it helps to know that the questionnaire itself was not particularly oriented towards the motivational strengths of computer games. Although there were some questions that related to the instruction's ability to hold one's attention, many of the questions inquired about the relevance of the instruction and the student's confidence that he or she had achieved mastery. *Electro Adventure*, by the way, fared particularly poorly on the confidence scale.

In understanding the results of the evaluation in general, it is helpful to know that Electro Adventure in its current version suffers from extensive technical flaws including software bugs, inaccurate technical content, flaws in instructional technique, and human-interface problems. Despite these problems it fared well on the single criterion that was most important to us as developers, and it did not do too badly on other criteria.

It also helps in understanding these results to take the proper perspective on the state of the art in this arena. *Electro Adventure* is, to my knowledge, the only computer game that seeks (with some success) to be a complete replacement for instructional techniques that have evolved over years in the particular school and millennia in general.

My ambitions for *Electro Adventure* were the same as those of the Wright brothers at Kitty Hawk, namely that it would fly. That it not only flew but also carried the freight was both gratifying and unexpected. The data in Table 1 constitute the first point of the learning curve for this very new technology. Although these data do not constitute a strong recommendation for *Electro Adventure* itself, they do indicate that investments in improving the technology would be well worth the expense.

HIGH TRANSFER TRAINING (HITT): INSTRUCTION DEVELOPMENT PROCEDURES AND IMPLEMENTATION STRATEGIES

Dorothy L. Finley and Michael G. Sanders
US Army Research Institute Field Unit
Fort Gordon, Georgia 30905

ABSTRACT

Some jobs, including many Army Military Occupational Specialties (MOSs), require the job incumbent to perform operations or maintenance tasks on several different objects (e.g., items of equipment, reporting forms) or different object configurations. It has often been observed that job incumbents in Army units resist performing tasks on equipments, for example, on which they say they have not been specifically trained. A problem, however, is that the training time needed to train persons on many different objects and in every configuration is prohibitively costly. And, given limited equipment availability, for example, such training may not be possible even if it were affordable.

HIgh Transfer Training (HITT) is a new methodology for developing training programs which can be applied when needed to resolve the problem of meeting training requirements for multiple sets of related objects. HITT is an extension of the Systems Approach to Training (SAT) in that it adds some analytical steps and training implementation strategies to SAT. The HITT analyses enable the training developer to identify and codify similarities between objects and between object configurations, and then to group them into families according to the similarities. The family groupings provide the basis for implementing training strategies which enable soldiers to transfer school knowledge to equipment sets and configurations which differ from those on which they received specific training.

The HITT methodology consists of a two-phased training development process, Task Generalization and Generic Design. The emphasis in this paper will be on Task Generalization phase and the HITT training strategies. This paper will also briefly recount the history of HITT and evidences of its value-added effectiveness.

ABOUT THE AUTHORS

Dorothy L. Finley is a Principal Scientist and former Team Leader. Her new address is: ARI Armored Forces Research Unit, ATTN: PERI-IK, Fort Knox, KY 40121-5620. Phone: (502) 624-7046.

Dr. Michael G. Sanders is now Chief of the ARI Fort Bragg Coordination Office. His new address is: SOFT, ATTN: Psychology Section/Dr. Mike Sanders, PO Box 70660, Fort Bragg, NC 28307-0660. Phone: (910) 396-0874.

HIGH TRANSFER TRAINING (HITT): INSTRUCTION DEVELOPMENT PROCEDURES AND IMPLEMENTATION STRATEGIES

Dorothy L. Finley and Michael G. Sanders
US Army Research Institute
Fort Gordon, Georgia

BACKGROUND

Statement of Problem

Some jobs, including many Army Military Occupational Specialties (MOSs), require the job incumbent to perform operations or maintenance tasks on several different items of equipment or different configurations of an item. It has often been observed that job incumbents in Army units resist performing tasks on equipments on which they say they have not been specifically trained. A problem, however, is that the training time needed to train persons on many different equipments and in every configuration is prohibitively costly. And, given limited equipment availability, such training may not be possible even if it were affordable.

In recent years, one approach to keeping training costs under control in these situations has been described as "generic" training. The concept is one of presenting training in a fashion that will result in effective transfer of training across a group of related equipments (Knipling, Ryan, & Sanders, 1989, p.2). It has been especially used in the electronics area (Casper, 1986; Knipling et al., 1989; Office Chief of Ordnance, 1986). In developing a "generic" course, some form of analyses has sometimes been performed using other previously developed analyses (e.g., task analyses) as inputs. Investigation determined that there were no standard **procedures** for development of generic training, no standard **form** of generic training, nor knowledge of the **extent** to which any of these programs were more successful than others.

The Army's current standard approach to training development is a modification of Instructional Systems Development (ISD) called the Systems Approach to Training (SAT) (Department of the Army (DA), 1988). It encompasses the concept of generic training in that skills and knowledges are to be taught such that they are generalized to related objects when such exist. The problem, however, has been lack of analytic procedures to implement that generic training concept (M. McAllister, personal communication, 20 May 1994).

The development and testing of the HIgh Transfer Training (HITT) methodology has been in response to these concerns. The HITT program objectives are to provide:

- more precise knowledge on **when** to implement generic training in order to realize **what** gain; and

- systematic and standardized procedures for developing it.

History

HITT extends the Army's SAT in that it adds some analytical steps and training implementation strategies to SAT. It should be noted that the required input to HITT, which can be provided by the early steps of SAT, is complete and accurate job and task information. HITT could, therefore, complement other training development procedures which similarly develop task and job information.

The first course developed through use of HITT is now operational at the U.S. Army Signal Center. That course is Advanced Individual Training (AIT) for the 31M MOS, Multichannel Transmission Systems Operator. Evaluations to validate and determine the value added by the use of HITT have been conducted and are briefly summarized below. It is the intent of the U.S. Army Training and Doctrine Command to incorporate HITT into the procedures supporting the new Army Training Development Model (M. McAllister, personnal communication, 20 May 1994). The draft regulation describing this model specifically defines generic training as "Training skills and knowledges that support a number of individual tasks" (DA, 1994, p. B-11). The new model introduces new terminology and expands, refines, and will replace such existing regulations as the ones covering SAT. The HITT procedures documentation is currently undergoing revision to reflect these changes.

Evaluations Have Established HITT's Value Added

The first investigations established that soldiers trained in the HITT-developed 31M course performed very well (Finley, Shipman, & Lowry, 1993). Their performance on the equipment items on which they had been specifically trained exceeded that of students trained under the previous 31M course. That course was designed using only the current SAT approach.

A subsequent investigation further established the value of the HITT approach when the performances of 31M HITT-trained students were compared to those of SAT-trained students in another MOS (Finley, Singleton, Drago, & Sanders, 1993). The performances of the students were compared on operations, preventive maintenance, and trouble shooting tasks on a major item of equipment on which none had trained and for which neither MOS was responsible. That equipment item had some components with various degrees of similarity to components on which both MOSs had been trained - and others on which neither MOS had been trained. Even though their aptitude scores were significantly lower - the HITT-trained soldiers performed significantly better on all tasks.

INSTRUCTION DEVELOPMENT PROCEDURES

Overview

A major step in developing programs of instruction is some form of job and task analysis. Early in the Army's current SAT process, often with subsequent reviews, decisions are made as to which specific tasks (i.e., specific action verbs and objects) are sufficiently critical to warrant the time and support costs of training (e.g., install a 3 KW gasoline generator set, complete maintenance report form XXX-Y). SAT then focuses the training developer's attention on performance within these specific tasks with their specific object.

As a hypothetical example, a block of instruction on generators developed using the SAT approach may, to some extent, break out platform instruction on generator principles as some holding true for all types of generators and others as applying only to subsets of generators. In any case the focus would be on a specific generator and the particular principles applying to it. Hands on skills training on more than one type of generator would not be provided to every student.

The HITT methodology extends SAT analyses in situations where the soldier must perform on a multiplicity of objects and/or object configurations. HITT enables the analyst to examine the entire set of tasks and objects for which the soldier is responsible and determine if certain commonalities exist between some of the tasks, particularly with respect to the objects of the verbs. To the extent commonalities exist, HITT then enables training developers to codify these similarities in a manner which allows to group the objects and/or object configurations into families of functionally associated members. Commonalities and differences between family members are defined down to the lowest levels of interface between the objects and the soldier (a driver versus a transmission mechanic, for example, have differing levels of machine interface with a jeep). The functional groupings provide the basis for incorporating certain training transfer strategies into the program of instruction (Neal, Lowry, Finley, Sanders, & Ryan, 1994).

The result of the differing perspective that HITT affords can be described by again using the above generator example. Assume again that installing generators was judged to be a critical responsibility for an MOS. With analyses, the HITT developer might find the 3 KW, 5KW, and 10 KW gasoline generators comprised one of the families of generators for this MOS. The students would not only be educated with respect to commonalities and differences between the 3 KW, 5 KW, and 10 KW gasoline generators, they would also all train on more than one under differing conditions. If another generator was also the responsibility of that MOS, but found to be quite unique, then education and training would also be provided on that generator (to the extent warranted) but it's relationship to the above family would be clearly articulated.

Procedures

The HITT methodology consists of a two-phased training development process, Task Generalization (Steps A - D) and Generic Design (Steps E - H). An overview of the use of HITT within the context of SAT or other training development procedure is presented in Figure 1. The reader should note in Figure 1 that a decision point is reached after completion of Step B in Phase I. Applying the first two steps is not labor intensive. If, in applying Steps A and B, functional commonalities are found then the training developer, using these commonalities, can proceed to complete the development of a program of instruction using the HITT methodology. If no commonalities are found then the training developer would complete the course design using SAT or other procedures. The difference is that if commonalities are found then these can be used in HITT Steps C - H to incorporate transfer training strategies into the design of the program of instruction.

Most of the attention here will be focused on Steps A and B. The transfer training strategies are discussed later.

<u>HITT Phase I</u> The Task Generalization process consists of four major steps:

A. Prepare Objects and Verbs Lists;

B. Develop Generalized Components and Objects Lists;

C. Describe Generic Action Statements; and

D. Describe Knowledges and Skills Groups associated with Generic Action Statements.

In Step A, the training analyst reviews the critical task list and the job task inventory to ensure that the critical task list is complete. From the critical task list, the analyst identifies the objects and actions for which the MOS is responsible. The analyst sorts the objects into functional groups. As examples of such groups, all receiver/transmitter units have the same general purpose, or function, and all reports have the same general function. If an object does not fall into any group having other objects as members then it is treated as an additional functional group that has only one member. All action verbs related to the objects are similarly sorted into functional groups of actions. As an example, the action verbs "remove and replace", "diagnose", "disassemble", "assemble", and "test" might be sorted into a group that is named "repair".

In Step B, the analyst works with the list of functional object groups developed in Step A. The purpose of Step B is to further definitize commonalities and differences between objects within and between the functional groups of objects. Based on the outcomes of these analyses, a Generalized Objects list is created which identifies these commonalities and differences.

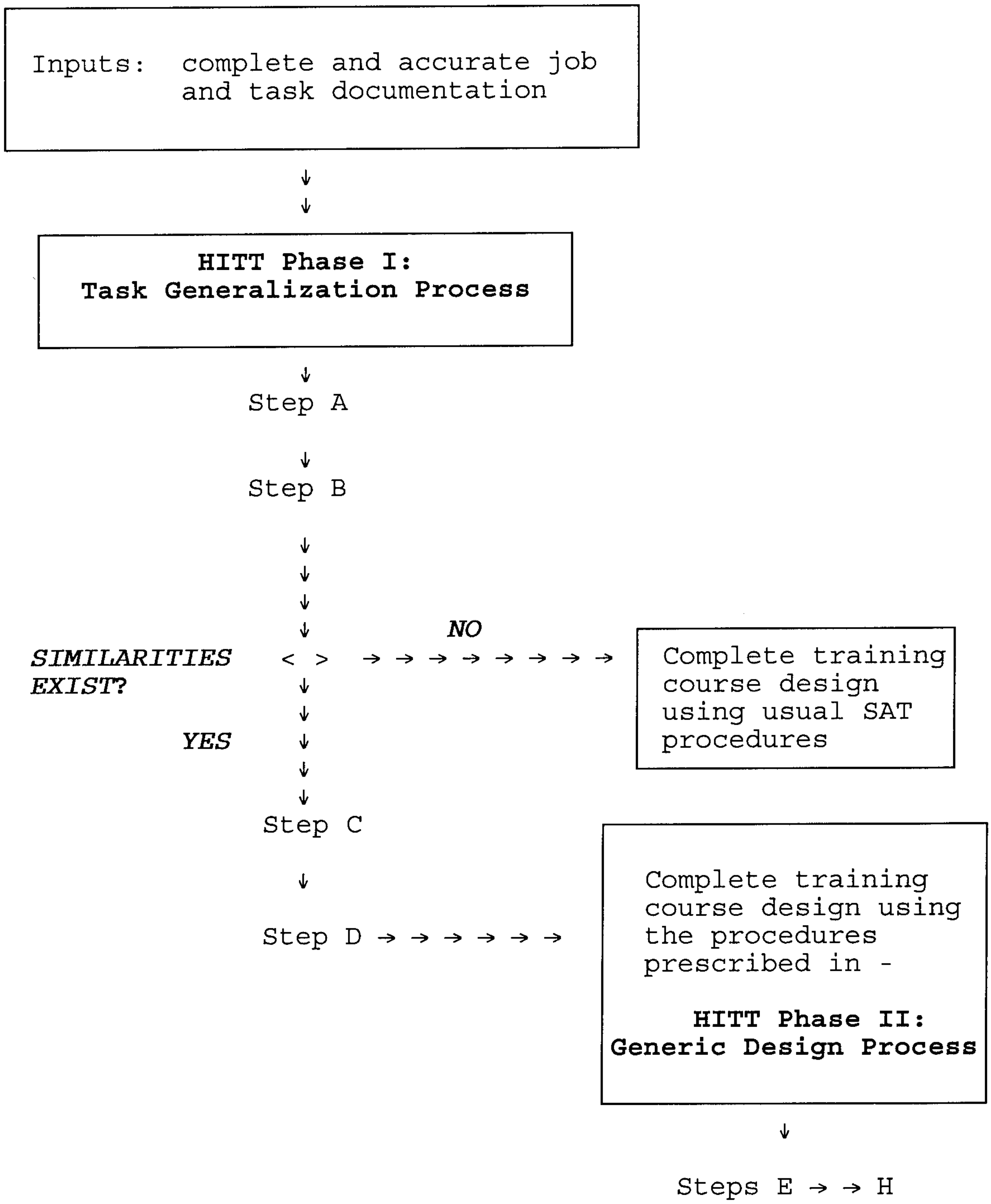


Figure 1. The integration of HITT into the training course development process.

The Generalized Objects list also includes objects determined to be unique from Step A and these are identified as such. Given this further definitization, some groups may be renamed and/or their membership may change. The analyses performed in Step B provide the information needed to determine whether or not - and how - HITT strategies can be implemented in course design.

Components are defined as parts of objects that serve a distinct purpose from the soldier's viewpoint and requires action on his/her part. The process for Step B is to identify and analyze components of the objects down to the lowest level of actual soldier-object interface. What is sought is a determination of which components are: (1) functionally and physically common; (2) functionally and operationally common; or (3) unique.

In Step C, the list of action verbs and their functional groupings, if any, from Step A is matched to the Generalized Objects List (including the unique objects) from Step B. The extent of the match between objects is then further defined by comparing the details of their task processes. The outcome is the creation of Generic and Specific Action Statements (GASs, SASs). The GASs are as generic as possible with any important differences in task process noted through associated SASs. Some SASs may be stand alone itemsbecause the objects and verbs were found to be truly unique.
Completion of Step D - describe knowledges and skills groups associated with the Generic Action Statements - provides the rest of the information needed to determine the resident, or other, training actions and to begin HITT Phase II - the instructional design process.

HITT Phase II The second HITT phase, the Generic Design Process, uses steps and methods similar to those used in the SAT process. The difference is that HITT additionally guides the integration of certain transfer training strategies into the course design. These strategies are discussed below. The four design steps are:

E. Develop and Sequence Generic Terminal Learning Objectives;

F. Develop Learning Specifications;

G. Develop Syllabus; and

H. Develop Lesson Materials.

IMPLEMENTATION STRATEGIES

The objectives of HITT implementation strategies are to teach the student: that certain similarities exist between at least some equipments, or other objects, in your MOS; how to recognize and respond to these similarities; that even though you were not trained on a specific item, you may still be able to operate and fix it; and that you do have resources - use them to figure out how to proceed. As described above, there is strong evidence that soldiers trained in this manner perform better. Observational evidence has been that HITT-trained soldiers also exibit much greater confidence and are eager - rather than reluctant - to work with equipment on which they have not been specifically trained.

Four transfer training strategies are implemented through use of HITT. They are:

(1) Training each student on three object and/or object configuration members from each GAS. One object is designated the base task, another is designated the intermediate task, and the last is designed the transfer task. They are trained in that order, using different procedures.

(2) Allowing students to interact and help each other while learning their intermediate tasks.

(3) Training the students to use their primary resource materials, e.g., their technical manuals (TMs) as a part of performing their jobs.

(4) Telling the students at the outset that they will be able to transfer what they have learned in school to their job in the unit and why.

With regard to the first two strategies, a class of students is separated into three groups. For each GAS, each group is assigned a base task, an intermediate task, and a transfer task. If the number of objects and/or object configurations within a GAS is more than three then the three objects assigned to each student group may be different. In any event, each group will differ in that the base task assigned to them, for example, will not have the same object as that assigned to either of the other two groups.

The base task is instructed to all students within a group, followed by instructor-guided hands-on practice. As a part of this, the instructor provides overall descriptions of the object, the functional group of which it is a member and the similarities between the group members, its functions, and the concept of operation. The students then attempt to learn on their own how to perform the intermediate task they have been assigned. The instructor provides assistance when requested and the students are allowed to discuss any problems with other students, whether in their group or not. (Although students in other groups will not be learning the same intermediate task for that GAS, the tasks will have similarities due to the fact that the HITT analysis grouped the objects into the same family). Finally - entirely on their own and at their own speed within a structured time frame - the students try to learn how to correctly perform their transfer task. Here, students are not allowed to ask for assistance and are evaluated when they feel ready to be graded.

With regard to the third transfer training strategy identified above, the students are trained to seek out and actively use their resource materials throughout all of the foregoing. It should be noted that this is not the usual strategy for non-HITT developed courses. Because it is easier for the instructors and creates fewer resource demands, students in most courses are given only references or handouts abstracted from the TMs by the teacher.

RESEARCH BASIS

The HITT approach is firmly based on extensive research on how to enhance transfer of training. (For summarizations see Adams, 1987; Cormier & Hagman, 1987; Druckman & Bjork, 1991, chap. 3; and Patrick, 1992, chap. 3). The relationships between HITT strategies and transfer of training principles are:

(1) Teaching the students to expect that there will be similarities as well as differences between objects they work with, what some of these are, and that this knowledge will transfer to what they find in their units (Noe & Schmitt, 1986; Dov & Shani, 1982).

(2) Training the students to perform on three different members of a functional group encourages the integration of similarities into the students' cognitive processes and formulation of a more abstract framework (Hagman, 1980; Druckman & Bjork, 1991; Patrick, 1992).

(3) Allowing groups of students to interact within and between groups, articulating questions and answers about related objects. This fosters a feeling of confidence, expectation and understanding of variety, and, again, a formulation of a more abstract framework (Ehrenber, 1983; Druckman & Bjork, 1991).

(4) Reducing the amount of feedback to students while training the second and third members of a functional group fosters an ability on the part of students to act independently (Druckman & Bjork, 1991).

(5) Training the students to actively use all resources available to them fosters a positive attitude regarding and ability to deal with somewhat different situations (Shields, Joyce, & VanWert, 1979; Adams, 1987)).

REFERENCES

Adams, J.A. (1987). Historical review and appraisal of research in the learning, retention, and transfer of human motor skills. Psychological Bulletin, 101(1), 41-74.

Casper, P. (1086). The generic training concept at the U.S. Army Signal Center (Technical Report). Fort Gordon, GA: U.S. Army Signal Center, Control and Specialized Electronics Department.

Cormier, S.M., & Hagman, J.D. (Eds.) (1987). Transfer of learning, contemporary research and applications. San Diego, CA: Academic Press, Inc.

Department of the Army (1988). Training: Systems Approach to Training (TRADOC Regulation 350-7). Fort Monroe, VA: U.S. Army Training and Doctrine Command.

Department of the Army (1994). Training Develoment Process, Management, and Product Development (Draft TRADOC Regulation 350-XX). Fort Monroe, VA: U.S. Army Training and Doctrine Command.

Dov, E., & Shani, A.B. (1982). Pygmalian goes to boot camp: expectancy, leadership, and trainee performance. Journal of Applied Psychology, 67(2), 194-199.

Druckman, D., & Bjork, R.A. (1991). In the mind's eye: enhancing human performance. Washington, D.C.: National Academy Press.

Ehrenber, L.M. (1983). How to ensure better transfer of learning. Training and Development Journal, Feb., 81-83.

Finley, D.L., Shipman, M.G., & Lowry, C.A. (1993). Evaluation of training outcomes resulting from use of the HIgh Transfer Training(HITT) methodology (Technical Report, in press). Alexandria, VA: U.S. Army Research Institute.

Finley, D.L., Singleton, J.M.H., Drago, J.W., & Sanders, M.G. (1993). HIgh Transfer Training (HITT) methodology: Further evidence of training benefits. Paper presented at the 37th Annual Meeting of the Human Factors and Ergonomics Society.

Hagman, J.D. (1980). Effects of training schedule and equipment variety on retention and transfer of maintenance skill (Research Report 1309). Alexandria, VA: U.S. Army Research Institute.

Knipling, R.R., Ryan, A.J., & Sanders, M.G. (1989). Generic training and its application to training communications systems operators. Paper presented at the Annual National Outreach in Technology and Education (NOTE) Conference.

Neal, J.D., Lowry, C.A., Finley, D.L., & Ryan, A.J. (1994). HIgh Transfer Training (HITT) methodology: Volume I. Procedures. Volume II. Examples (Research Products). Alexandria, VA: U.S. Army Research Institute.

Noe, R.A., & Schmitt, N. (1986). The influence of trainee attitudes on training effectiveness: Test of a model. Personnel Psychologist, (39), 497-523.

Office Chief of Ordnance (1986). Electronics maintenance structure (ELMS) study (Draft Final Report, ACN 082570). Aberdeen Proving Ground, MD: U.S. Army Ordnance Center and School.

Patrick, J. (1992). Training: research and practice. New York, NY: Academic Press.

Shields, J.L., Joyce, R.P., & VanWert, J.R. (1979). Chaparral skill retention (Research Report 1205). Alexandria, VA: U.S. Army Research Institute.

INSTRUCTIONAL DESIGN: INTEGRATION OF COGNITIVE STYLE AND TECHNICAL CONTENT

Linda J. Brent and Richard P. Brent
Loral Defense Systems
Akron, Ohio

ABSTRACT

Basic research in neuropsychology, learning theory, and cognitive psychology have contributed to knowledge concerning human learning. This research has been applied to the identification of cognitive styles -- defined as an individual's unique method of processing information. Investigations into ways to apply this knowledge through computer-based instruction, the increased use of multimedia technologies, and the integration of artificial intelligence techniques have enhanced occasions for more effective use of computer-based instruction in training applications. While technological advances permit more cost-effective solutions for individualized training, instructional designers may lack adequate techniques for integrating the advances in learning theory and cognitive style with the technology.

The current research literature acknowledges the importance of accounting for the nature of the subject-matter content. Guidelines concerning information presentation in computer-based instruction are needed by instructional designers to accommodate the individual cognitive style of the learner and for the differences in presentation format relative to subject-matter content.

This paper reviews current research, and discusses how instructional designers can integrate the research findings into a paradigm for the effective instructional design of interactive computer-based instruction. The paper describes appropriate design strategies which integrate the application of cognitive style research findings with subject matter content and multimedia capabilities. Specific examples of situations, learning scenarios, and strategies are provided. Directions for future research are also presented.

ABOUT THE AUTHORS

Linda J. Brent is currently the Training Manager for a major AF program, responsible for the integration of training and design requirements for all media developed. Since joining Loral in 1991, she has provided technical leadership for the conduct of two AF schools, curriculum development, and for training/ engineering interface in the final design and production of the training media. Telephone (216) 796-5913

Richard P. Brent is currently responsible for providing technical leadership for engineering process improvement initiatives, especially in systems engineering. Since joining Simulation and Training Systems Engineering in 1993, he has conducted requirements analysis and feasibility studies, defined optimal training system configurations, and developed technical plans for training system development.

INSTRUCTIONAL DESIGN: INTEGRATION OF COGNITIVE STYLE AND TECHNICAL CONTENT

Linda J. Brent and Richard P. Brent
Loral Defense Systems-Akron
Akron, Ohio

INTRODUCTION

Basic research in the areas of learning theory, memory, neuropsychology (hemisphericity), and cognitive psychology (information processing) has made significant contributions to the body of knowledge concerning how human learning occurs. This knowledge has been applied to the identification of cognitive styles. Cognitive style, initially termed such by Allport (1937), has been described as an individual's typical mode of thinking, problem-solving, perceiving, and remembering (Schwen, Bedner & Hodson, 1979). Ausburn and Ausburn (1978) referred to cognitive style as the psychological dimensions that represent consistencies in an individual's method of acquiring and processing information. Research into ways to apply this knowledge, including technological developments in the use of computer-based interactive instruction and the integration of artificial intelligence techniques can help establish more effective and cost-efficient use of computer-based instruction in a wide variety of training applications.

Instructional designers, however, may not currently have adequate tools and techniques for the development of instructional programs tailored specifically to the individual needs of the learner. In many of the current military training programs, it is assumed that all learners process and store information in the same manner. Recent research suggests that learners absorb, process, and act on information differently (Corno & Snow, 1986; Kogan, 1971; Messick, 1984). For example, some individuals best retain information presented graphically and holistically, whereas others best process information serially, with verbal presentation. Some researchers (Riding & Sadler-Smith, 1992) also acknowledge the importance of taking into account the nature of the technical content or task characteristics when planning the best instructional approach. This can facilitate retention and eventual transfer of training to the job situation.

In addition to the cognitive style characteristics of learners, designers of interactive instruction may not have adequate guidelines for the development of programs which consider the individual cognitive style characteristics of trainees and the most effective mode of presentation for the particular technical content requirements of the material. The problem here lies in the lack of integration of cognitive style factors and the critical content elements which interact and affect learning, retention, and training transfer. If the elements of technical content, cognitive style (both process and representation of information) are consistently considered and applied when designing training, the effectiveness and efficiency of learning can be enhanced.

We shall review the current research in the areas of cognitive style, instructional format and design and provide guidelines for the design of interactive instruction (i.e., interactive courseware, [ICW]).

RESEARCH IN COGNITIVE PSYCHOLOGY

Research in neuropsychology supports the notion of a lateralization of functions in the two hemispheres of the brain, with the right hemisphere predominantly responsible for spatial, holistic, inductive processing and the left hemisphere predominantly responsible for analytic, sequential, and verbal processing (Messick, 1966; Wittrock, 1980). However, more recent research suggests that the cooperative processing of both hemispheres are required for the most efficient synthesis of information.

Hemispheric processing is considered a continuum in which dominance is distributed. The utilization of both hemispheres for certain tasks has been demonstrated, but differential aptitudes in functions may lead to the emphasis of one hemisphere over another in a particular individual's processing mode (Dumas & Morgan, 1975). The results of studies in this area suggest that individuals possess preferred processing modes, but with increased skill or complexity, shared hemispheric processing responsibilities increase (Kolers & Roediger, 1984). There is also evidence to suggest that although information is processed with both hemispheres, individuals tend to process information differently (Kagan, 1965; Pelligrino & Kail, 1982). These differences have been further expanded and addressed in studies which focus on the area known as cognitive style.

Cognitive style has been defined as ". . . self consistent, enduring individual differences in cognitive organization and function." (Ausubel, Novak, & Hanesian, 1978, p. 203). Cognitive style is thought to include all processes used in the processing of information: perception, thought, memory, imagery, and problem solving. These individual cognitive styles appear to be related to hemispheric dominance (Wittrock, 1978) and differences in the modes of processing information (Ausburn & Ausburn, 1978). These differences are not related to which hemisphere is used, only to the degree to which one is used over the other.

Research on cognitive style has presented various models which describe these dichotomous style characteristics. For example, Witkin's (1965) research defined individual differences as related to the characteristics of field independence and field dependence, based on the cognitive perception of information received. Field-independent individuals perceive information analytically and field-dependent individuals perceive globally. Pask and Scott (1972) divide cognitive style into two primary modes according to the processing of information. The first, serialist mode individual, prefers to process information in a progressive, developmental, sequential pattern, while the second, the holistic style individual, prefers processing in a more global perspective. Lohman (1979) also conducted research into the cognitive processing of information based on visual/spatial and verbal inputs.

Some researchers have argued that the dichotomous identification of cognitive style using differing labels are indeed different conceptions of the same style dimensions (Miller, 1987). By combining the results of research on cognitive style, some general characteristics emerge. First, it appears that an individual's cognitive style remains stable over time and across tasks (Riding & Sadler-Smith, 1992). Second, the relationships between various style dimensions appear to hold true, and have similar dichotomous characteristics. For example, holistic processing, inductive reasoning, and field dependence appear to be related to each other and to right-hemispheric processing (Ausburn & Ausburn, 1978; Riding & Cheema, 1991). Third, there is evidence to suggest a relationship between cognitive style and particular learning tasks or modes of information representation (Kozlowski & Bryant, 1977).

Buehner Brent (1987) presented a theoretical framework for grouping the dichotomous cognitive style relationships related to hemispheric processing (Table 1). Riding and Sadler-Smith (1992) have synthesized the research on cognitive style and have grouped cognitive style dimensions into two dichotomous families. These families, the wholist-analytic and the verbal-imagery, address both the representation/perception and the processing of information. The first, the wholist-analytic dimension of style, describes the preferred manner in which individuals cognitively process information, in whole or in parts. The verbal-imagery dimension of style describes the preferred mode of representation of information, by verbal associations, or by the development of mental images, pictures, and associations. An individual's cognitive preference on each dimension is considered independent of one another. For instance, one individual's preferred style may be characteristic of a wholist and an imager, and another individual's style may be characteristic of an analytic and an imager. Individuals fall along the continuum of each style dimension.

When processing information as described in the wholistic-analytic cognitive style dimension, the wholistic thinker appreciates the overall perspective in the total context. In a learning situation, however, the same individual will require assistance in understanding the components which provide the structure for the task. The individual with the analytic style, on the other hand, will initially perceive the components that are to be learned, but will require an overview or more global perspective in order to successfully integrate the respective sections into a whole. This knowledge of an individual's learning style can help to define valuable design criteria for instructional material.

Likewise, the second style dimension of verbal imagery identifies an individual's preferred mode of receiving information. While the imagers learn best

Table 1 - Cognitive Style Dimensions

Left-Hemisphere Dominant	Right-Hemisphere Dominant	Relevant Studies
Field-independence	Field-dependence	(Witkin, Moore, Goodenough, & Cox, 1977)
Reflective	Impulsive	(Kagan, 1965)
Serialist	Holist	(Pask, 1972)
Analytic	Wholist	(Riding & Sadler-Smith, 1992)
Verbal	Visual	(Riding & Sadler-Smith, 1992)

from simple, graphic, pictorial representations or explanations which are easily visualized, verbalizers prefer primarily textual/verbal explanations of the content or graphics laden with detailed verbal information. An understanding of this dimension of cognitive style, by instructional designers, who also consider the way information is represented, can help to design interactive instruction with flexibility in form and format for students.

When a task requires a transformation in processing that is incompatible with a learner's style (for example, purely visual information for a verbal style), the learner may not perform the task successfully. Therefore, instructional designers should consider these styles as a factor when planning instruction, particularly interactive instruction. Ineffective or inefficient learning of technical information may not always be learner-based (e.g., lack of ability). Instead, it may be considered instructional-based, if there was no consideration in the design process for individual cognitive style factors and the nature of the technical content. If one accepts the postulate that cognitive styles and effective learning is an instructional rather than a learner-based issue, it becomes important to devise a means by which instructional modification can be successfully accomplished.

COGNITIVE STYLES AND INSTRUCTIONAL DESIGN

The identification of strategies for presenting information which link cognitive style variables and educational applications has been described by many researchers. Miller (1980), for example, stated that the instructional designer's role is to devise conditions in the learner's external environment which support the learner's internal cognitive process. Others (Frederico & Landis, 1984) state that consideration of cognitive style dimensions can aid individuals in learning information more readily and in retaining/retrieving information more effectively. Grasha (1984), while supporting the need for a match between style and information mode of presentation, also emphasizes that too consistent a match could create a non-motivational attitude in learning, by not encouraging accommodation to variety. The model presented by Brent (1987, 1990) supports the identification of preferred style by the development and use of both hemispheres in processing through the mode of hierarchical processing within cognitive style. The previous discussion concerning the efficient use of processing across the two hemispheres of the brain supports this view as well.

Several cognitive style dimensions appear especially suited for consideration in the design of training programs. For example, in considering the wholistic-analytic style dimension family, the format or structure of the material to be presented could be critical in successful learning. Likewise, in presenting information in a combination of textual and graphic form, an understanding of the verbal-imagery dimension is also relevant. Several research studies have supported the relevance of cognitive style considerations in design. For example, Riding and Douglas (1992) found that for the verbal-imagery cognitive style dimension, the method of information presentation affects learning. While imagers seem to learn best from pictorial representations, verbalizers learn best from verbal presentations. In a study conducted by Brent (1990), the format of instructional material influenced performance, and the most effective presentation format was dependent on the nature of the technical content (e.g., a knowledge, skill, attitude, or decision-making task).

A study conducted by Geiselman and Samet (1982) demonstrated that learning performance was enhanced when subjects were permitted to organize and format information to meet their individual styles. Most technical content tasks will require some differing degree of organization and varying forms of representation, depending on the cognitive style variations among learners across the style dimensions. The format and structure of learning material will affect learners differently. For example, those with the style of verbal-analytic can increase learning when provided with a learning situation emphasizing discrete elements of the content to be learned. The wholist-imager will understand wholes and use diagrams depicting the whole content task. Instructional practice, then, should provide information in a form requiring both the identification of individual elements and the integration into the whole concept to be learned.

Several researchers have found that accommodating learner cognitive style preferences in an instructional presentation can lead to increased achievement and retention of the content material. Napolitano (1986) found that learning was enhanced by using instructional design strategies and formats of information which were congruent with their diagnosed learning styles. Dunn, Deckinger, Withers, and Katzenstein (1990) also found increased achievement among students when information was presented in a form and format sensitive to their cognitive style. McRobbie (1991) found that dimensions of cognitive style were diagnostic predictors for both cognitive organization and knowledge outcome. The findings in his study demonstrate the importance of the cognitive style in the area of science learning. This is based on the learner's knowledge of his/her own style, and the appropriate strategies to employ (e.g., presentation and organization of information) in order to maximize learning.

Considering the research on cognitive style and its interaction with instructional format and content, several general principles emerge which can be applied to instruction and training. These principles are:

1. Individuals vary in the way in which they most effectively process/retain information.

2. The cognitive styles of individuals differ in relation to their hemispheric dominance for information processing.

3. The most efficient, effective learning occurs when processing of information involves use of both hemispheres, and the design of instruction encourages this processing to occur.

4. Instructional design features sensitive to cognitive style variables should consistently include:

   a. Opportunities for the learner to adjust aspects of the instructional environment (e.g., organization of the content, perspective, etc.).

   b. A combination of verbal and spatial information, closely related to one another.

   c. A flexible instructional organization, which includes an overview in verbal and spatial format, and opportunities for review and reinforcement.

   d. Opportunities for the learner to apply new information to a variety of learning situations.

5. Instruction should remain consistent with an individual's cognitive style.

Using these principles in the design of instruction may create unique problems for the development of instructor-led training. However, the increased use of computer-based interactive instruction (ICW) in training programs provides tremendous opportunities to individualize training which is sensitive to cognitive style dimensions. Interactive instruction provides opportunities for graphics, animation, video, and sound, along with the capability to branch and provide student feedback in a variety of ways. As discussed in AF Handbook 36-2235, (Department of the Air Force, 1993) on the instructional design process, interactive instruction has the capability to provide flexibility for the learner by presenting information in visual or verbal formats, with student interaction, error response capability, and animation.

A significant trend in the military is the increased use of computer-based interactive instruction. In many such programs to date, however, the learner must adapt to the format of the instruction provided. The capabilities of the computer-based system are not used to take advantage of the flexibility in presentation. The exciting potential of this media, however, is in its capability to present individualized instruction, designed to be modifiable to meet the needs of all learners in a variety of learner settings and situations. Additionally, the capabilities of new processing and development systems like Hypercard create even more opportunity for flexibility, especially where learning situations are not purely linear in nature. The key to effective use of this media to maximize learning is to develop instruction which appropriately uses graphics, text, animation, and so forth, which is based on cognitive style dimensions and on the nature of the content.

IMPLICATIONS OF RESEARCH ON INTERACTIVE INSTRUCTIONAL DESIGN

As discussed, current research supports the integration of cognitive styles and technical content. With the flexibility provided by interactive computer-based instruction, effective and efficient learning can be enhanced. General principles emerging from the research in this area can serve as general guidelines in the design of learner-based training. These guidelines are as follows:

1. Cognitive style research useful to instructional designers can be synthesized into two dichotomous style dimension families: wholist-analytic, and verbal-imagery. These two dimensions describe the way individuals process/organize information and the way in which they represent the information.

   Design Principle for Wholist-Analytic Dimension: Provide opportunities for learners to format information in a variety of organizational formats. For example, provide an advanced organizer at the beginning of a lesson to define the overall lesson purpose, its components, and the way in which the portions of technical content within the lesson fit together. Permit opportunities for the learner to access this information throughout the lesson in order to put the detailed information being presented in a more global context.

   Design Principle for Verbal-Imagery Dimension: Provide information in at least two forms, verbally (in text) and visually (in simple pictures/diagrams). For example, present the advanced organizer described above both verbally (in text) and visually (in a simple hierarchical diagram). Permit the learner flexibility to access the preferred format of information throughout the lesson.

2. Cognitive styles are resistant to change. As such, the instructional designers should not expect the learner to adapt his individual style to the content and format of instruction. Instead, the instructional format should be designed to maximize learning across style dimensions.

   Design Principle for Wholist-Analytic Dimension: Provide cues to the learner throughout the lesson which relate the content presented as part of the entire lesson material. For example, if the purpose of the lesson is to define a series of menu screens on an aircraft multi-function display (MFD), design the lesson to permit the learner to return to a pictorial representation of the overall menu screen architecture.

   Design Principle for Verbal-Imagery Dimension: Provide both verbal and visual information whenever possible to instruct the technical content. In the case of the MFD example, use a windowing capability in the lesson to provide verbal information and direction to the learner while maintaining the graphic image of the MFD on the screen for reference and visual matching to the verbal information.

3. Although individuals tend to demonstrate preference for a given cognitive style, instruction should provide variety to challenge and integrate both hemispheres of the brain, enabling cooperation among style characteristics and maximizing the efficiency of learning.

   Design Principle for Wholist-Analytic Dimension: Use techniques such as color, highlighting, and windowing to define the learner focus on a given portion of the technical content. For example, use color coding of information to guide the learner in parsing the technical content into appropriate categories as to the way the information of the moment fits into the overall content being taught.

   Design Principle for Verbal-Imagery Dimension: Use a variety of color, animation, graphics and video inputs throughout the course of the lesson as appropriate for describing the technical content. For example, to demonstrate the loadmaster tie-down procedures of certain cargo, provide a written checklist, along with short video segments, to describe the steps in the checklist procedures.

4. While verbalizers can process detailed, complex information without the use of visual images or organizers, imagers process information best when it is presented in simple, concrete diagrams and visual images.

   Design Principle for Wholist-Analytic Dimension: Use the windowing capability of the technology to allow learner choice in defining the information needed at any point in time. The learner should be able to choose the information present at a given time rather than having a multitude of technical content information on the screen at any one time, which would require cognitive sorting and organization of unnecessary information.

   Design Principle for Verbal-Imagery Dimension: Provide information on the screen using the simplest visual format possible. Use simple graphics, animation and video to visually illustrate a technical concept, without making the presentation confusing by overloading the screen with superfluous information. For example, a simple line drawing may better demonstrate the concept of terrain following than a complex motion video of the out-the-window view from a plane engaged in terrain following. The complex video could prevent the learner from attending to critical elements in the description.

5. Provide flexibility for the learner to progress through a lesson by providing opportunities to individualize the instruction to match the cognitive style characteristics of the learner.

   Design Principle for Wholist-Analytic Dimension: Permit the learner to move freely back and forth through the lesson, with an ability to access frequently used concepts through a help menu system. For example, permit the learner to return to the overall advanced organizer of the lesson as required to visualize how the content being displayed at the moment fits into the overall concept being taught.

   Design Principle for Verbal-Imagery Dimension: At key instructional points, permit the learner to access a second format by which to demonstrate the information being taught. For example, if the technical content describes the characteristics of an aircraft system, provide opportunities for the learner to access wire diagrams, online which can enhance understanding of the operation of the system. For example, the use of Hypercard techniques provides the learner with the capability to explore technical content in this associative, nonlinear manner. It enables learners to make their own individual choices and follow their own learning paths.

6. Design the interactive instruction according to the type of information presented. Staver (1984) and Trafton (1984) recognized that learning was most effective when considerations of the type of technical content were made. This perspective is also supported by Brent's (1990) research where two content areas, particularly skills and decision-making, were impacted by instructional format, namely, visual/spatial information combined with minimal text.

   Design Principle for Wholist-Analytic Dimension: Use a visual web to define relationships among components of information. This technique provides structure for the technical content by providing a visual image of the interrelationships of the concepts.

   Design Principle for Verbal-Imagery Dimension: Design interactive computer-based instruction to provide opportunities for the learner to use a combination of verbal and visual information. The most effective presentation order for information is either to present them in synchrony, or present the visual before the verbal (Baggett & Ehrenfeucht, 1983). Further, when both are presented in synchrony, it may be advantageous to place visual information to the left of the verbal information to maximize processing by the right hemisphere of the brain (Wickens, 1984).

Today's economic climate dictates that training programs must be designed in a cost-effective manner. Using multimedia technology and interactive computer-based design techniques, instructional designers can consider cognitive style characteristics and instructional format considerations. If these methods are employed to enhance cooperative process, learners may obtain long-term retention and transfer of information benefits. Use of these design principles, over time, could reduce the length of training, improve its effectiveness, and therefore reduce its cost.

Future research should continue to focus on the use of interactive technologies to design instruction which integrate cognitive style and technical content dimensions. First, research should continue efforts to refine the critical elements of cognitive style. Research into the nature of these characteristics and their relevance to adaptive training models should be conducted.

Second, research should focus on the development of a computer-based cognitive style assessment tool which is based on the nature of content to be presented during the training course. This instrument could assess the most salient factors for effective performance in training tasks in the content areas of knowledge, skill, attitude, and decision-making. Research to date has identified several possible factors important to consider in the development of training programs (e.g., presentation format, cognitive style). With additional research, it may be possible to develop either a cross-content or content-dependent instrument to assess learner characteristics before training begins. Since computer-based training offers opportunities for the development of adaptive training programs, these programs could then be adapted to meet individual training needs.

REFERENCES

Allport, G. W. (1937). Personality, a psychological interpretation. New York: Holt and Company.

Ausburn, L. J. & Ausburn, F. B. (1978). Cognitive styles: Implications for instructional design. Educational Communications and Technology Journal (formerly AV Communication Review), 26(4), 337-354.

Ausubel, D. P.; Novak, J. D., & Hanesian, H. (1978). Educational Psychology: A cognitive view (2nd ed.). New York: Holt, Rinehart and Winston.

Baggett, P., & Ehrenfeucht, A. (1983) Encoding and retaining information in the visuals and verbals of an educational movie. Educational Communication and Technology Journal, 31 (1), 23-32.

Brent, L. J. B. (1990). Computer-based instruction: Effect of cognitive style, instructional format, and subject matter content on learning. (Technical Report AFHRL-TR-88-63). Brooks AFB, Texas: Air Force Systems Command.

Buehner Brent, L. J. (1987). Instructional design: Impact of subject matter and cognitive styles (AFHRL-TP-86-11, AD-A177 066). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

Corno, L., & Snow, R. E. (1986). Adapting teaching to individual differences among learners. In M. Wittrock (Ed.) Handbook of research and teaching (3rd ed.). London: MacMillan.

Department of the Air Force (1993). Information for Designers of Instructional Systems, Volume 5 (AF Handbook 36-2235). Department of the Air Force.

Dumas, R., & Morgan, A. (1975). EEG asymmetry as a function of occupation, task, and task difficulty. Neuropsychologia, 13, 219-228.

Dunn, R., Deckinger, E. L., Withers, P. & Katzenstein, H. (1990). Should college students be taught how to do homework? The effects of studying marketing through individual perceptual strengths. Illinois School Research and Development Journal, 26 (2), 96-113.

Federico, P. A., & Landis, D. B. (1984). Cognitive styles, abilities, and aptitudes: Are they dependent or independent? Contemporary Educational Psychology, 9 (2), 146-161.

Geiselman, R. E., & Samet, M. G. (1982). Personalized versus fixed formats for computer-displayed intelligence messages. IEEE Transactions on Systems, Man, and Cybernetics, SME-12 (4), 490-495.

Grasha, A. F. (1984). Learning styles: The journey from Greenwich Observatory to the college classroom. Improving College and University Teaching, 32 (1), 46-53.

Kagan, J. (1965). Individual differences in the resolution of response uncertainty. Journal of Personality and Social Psychology, 2, 154-160.

Kogan, N. (1971). Educational implications of cognitive styles. In G.S. Lesser (Ed.), Psychology and Educational Practice (pp. 242-292). Glenview, IL: Scott Foresman and Co.

Kolers, P. A. & Roediger, H. L. (1984). Procedures of mind. Journal of Verbal Learning and Verbal Behavior, 23 (4), 425-429.

Kozlowski, L. T., & Bryant, K. J. (1977). Sense of direction, spatial orientation, and cognitive maps. Journal of Experimental Psychology: Human Perception and Performance, 3, 590-598.

Lohman, D. F. (October, 1979). Spatial ability: A review and reanalysis of the correlational literature (Technical Report No. 8) Arlington, VA: Psychological Services Division, Office of Naval Research.

McRobbie, C. J. (1991). Cognitive styles and cognitive structure. Science Education, 75 (2), 231-242.

Messick, S. (1966). The criterion problem in the evaluation of instruction: Assessing possible, not just intended outcomes. Princeton, NJ: Educational Testing Service.

Messick, S. (1984). The nature of cognitive styles: problems and promise in educational practice. Educational Psychologist, 19, 59-75.

Miller, A. (1987). Cognitive styles: an integrated model, Educational Psychology, 7, 251-263.

Miller, G. G. (1980). The relevance of cognitive psychology to instructional technology. Proceedings of the 2nd Interservice/Industry Training Equipment Conference (Report No. Ad-A107437). Alexandria, VA: Defense Technical Information Center.

Napolitano, R. A. (1986). An experimental investigation of the relationships among achievement, attitude scores, and traditionally, marginally, and underprepared college students enrolled in an introductory psychology course. (Doctoral dissertation, St. John's University, 1986). Dissertation Abstracts International, 47 435A.

Pask, G. S. & Scott, B. C. E. (1972). Learning strategies and individual competence. International Journal of Man-Machine Studies, 4, 217-253.

Pellegrino, J. W., & Kail, Jr., R. (1982). Process analyses of spatial aptitude. In R.J. Sternberg (Ed.) Advances in the Psychology of Human Intelligence. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers, 311-365.

Riding, R. J. & Cheema, I. (1991). Cognitive styles - an overview and integration. Educational Psychology, 11, 193-215.

Riding, R. J. & Douglas, G. (1992). Performance on, and Attitudes to, Computer Based Training within Service Industries. (Sheffield, Learning Technologies Unit, Department of Employment).

Riding, R. & Sadler-Smith, E. (1992). Type of instructional material, cognitive style and learning performance. Educational Studies, 18 (3), 323-240.

Schwen, T. M., Bedner, A. K., & Hodson, K. (1979). Cognitive styles: Boon or bane? Viewpoints in Teaching and Learning, 55, 49-63.

Staver, J. R. (1984). Effects of method and format on subjects' responses to a control of variables reasoning problem. Journal of Research in Science Teaching, 21 (5), 517-526.

Trafton, P. R. (1984). Toward more effective, efficient instruction in mathematics. Elementary School Journal, 84 (5), 514-528.

Wickens, C. D. (1984). The multiple resources model of human performance: Implications for display design. AGARD/NATO Proceedings, Williamsburg, VA: 17-1 - 17-7.

Witkin, H. A. (1965). Psychological Differentiation and Forms of Pathology. Journal of Abnormal Psychology, 70, 317-336.

Witkin, H. A., Moore, C. A., Goodenough, D.R., & Cox, P.W. (1977). Field dependent and field independent cognitive styles and their educational implications. Review of Educational Research, 47, 1-64.

Wittrock, M. C. (1978). Education and the cognitive processes of the brain. In J. Chall and A. Mirsky (Eds.) Education and the Brain. Chicago: University of Chicago Press.

Wittrock, M. C. (1980). The Brain and Psychology. New York: Academic Press.

IMPACT OF TOTAL TRAINING SYSTEM ACQUISITION ON INSTRUCTIONAL SYSTEM DEVELOPMENT

Conrad G. Bills
Loral Defense Systems
Akron, Ohio

ABSTRACT

Traditionally, the Instructional System Development (ISD) process was applied for creating instruction in a classroom or a learning center. Historically, ISD grew out of a systems engineering methodology applied in development of self-paced programmed instruction. Successful programmed instruction resulted from a systematic development process. The system engineering concept provided a model for input, output, process, and feedback loop. The application of ISD to total training system acquisition brought a new perspective, the instructional system infrastructure. Top-level training system functions were identified and included in the systematic development model. The analysis phase was expanded to include functional analysis. The process for designing to function incorporated tools used in total quality management. This paper presents the impact of the expanded total training system perspective on the ISD process.

BIOGRAPHICAL SKETCH

Conrad G. Bills is an instructional technologist for the Simulation and Training Engineering Division, Loral Defense Systems, 1210 Massillon Road, Akron, OH 44315-0001; (216) 796-8705; FAX -7709. He had twenty years as U.S. Air Force officer.

IMPACT OF TOTAL TRAINING SYSTEM ACQUISITION ON INSTRUCTIONAL SYSTEM DEVELOPMENT

Conrad G. Bills
Loral Defense Systems
Akron, Ohio

INTRODUCTION

The step up to total training system acquisition by the Air Force caused a significant learning curve in the application of the traditional instructional system development (ISD) process. A process used for development of instructional materials was being driven to take into account a breadth previously outside the bounds of the instructional development team. The initial attempts to define this shift through the C-130 Model Aircrew Training System Study (Fishburne, Williams, Chatt, & Spears, 1987) were at first recognized and then set aside because these insights were not incorporated in the "official" publications. The resulting conflict was the force fit of a model designed for development of instructional materials into a role for which it was not designed.

The Advanced Aircrew Training System Study for United Airlines (Williams, Degen, Haskell, & Schutt, 1987) pointed out that in addition to the functions applied during the phases of the ISD process, there were the functions of management, support, delivery, and administration necessary for successful training system implementation. The ISD model needed to be updated to address these new perspectives required of it.

This new role for instructional system development (ISD) required integration of two perspectives. The first was the traditional focus on development of instructional content for delivery. The second was the expanded system perspective which focused on the management, administration, and support for delivery of the instructional content. Combining these two perspectives allowed focus on the broader "big picture" during the complex acquisition of the total training or educational system. Beyond implementation, these perspectives continue to bring quality improvement in day-to-day operations. A comprehensive analysis of the application of ISD in major aircrew training system acquisitions led to a new volume titled, <u>Information for Designers of Instructional Systems: Application to Acquisition</u> (AF HANDBOOK 36-2235, Vol. 3). The analysis for this volume showed that in acquisition, the ISD process must also interface with the systems engineering process.

The purpose of this paper is to summarize the impact of total training system acquisition on the ISD process. This summary includes approaches incorporated to meet the requirements of this expanded role. A number of these approaches again have their origin in the system engineering design process. The drawing together of terminology common to both processes facilitated communications in integrated product development.

Instruction

The first or traditional perspective of the ISD process is well established. The pattern for successful instruction, particularly self-paced instruction, is founded in a systematic development process (AFM 36-2234, 1993; Dick & Carey, 1990; Gagne, Briggs & Wager, 1974; Merrill, Lee, & Jones, 1990; Reigeluth, 1983; Tennyson & Michaels, 1991). This systematic development process usually takes place in the following phases: analysis, design, development, and implementation (AFM 36-2234). The role of evaluation has been expanded to apply within each of these phases, evaluation activities which are both qualitative and quantitative. These evaluation activities provide feedback about the quality of the process being applied and the products being developed. Steps are taken throughout the process to ensure continuous quality improvement, concepts adopted from total quality management.

Instructional System

The more contemporary perspective for the ISD process is an expanded total instructional systems perspective. In addition to the functions in each phase of the ISD process, success of the total training system is founded in implementation of the following basic functions: management, support, administration and delivery (AFM 36-2234, 1993; Bills & Butterbrodt, 1992; Williams, Judd, Degen, Haskell, & Schutt, 1987). Again, the expanded role of evaluation includes evaluation methods within each function for feedback about the quality of the activities being carried out. This pattern is true whether the setting is in training or in education.

The updated Air Force ISD Model shown in Figure 1 incorporates both the traditional and the contemporary perspectives for the ISD process in an overall model. When this general model is applied to specific settings, then prescription of the process is made to fit the application. Both the instructional content and the expanded instructional system perspectives are critical in the application to acquisition of total instructional systems. The ISD management plan should address both. The guidelines for specifications from systems engineering already do address both the system level or functional design and the allocation of functions to the follow-on component level or prime item design.

AREAS OF IMPACT

Golas & Shriver (1991) reported in the Baseline Analysis Report for the revision of the Air Force ISD process that ISD regulations are not adaptable to full spectrum training systems acquired as part of a major weapon system procurement. They also identified the need for an understanding between ISD and systems engineering personnel, such as relating corresponding vocabularies to development activities carried out by integrated product teams.

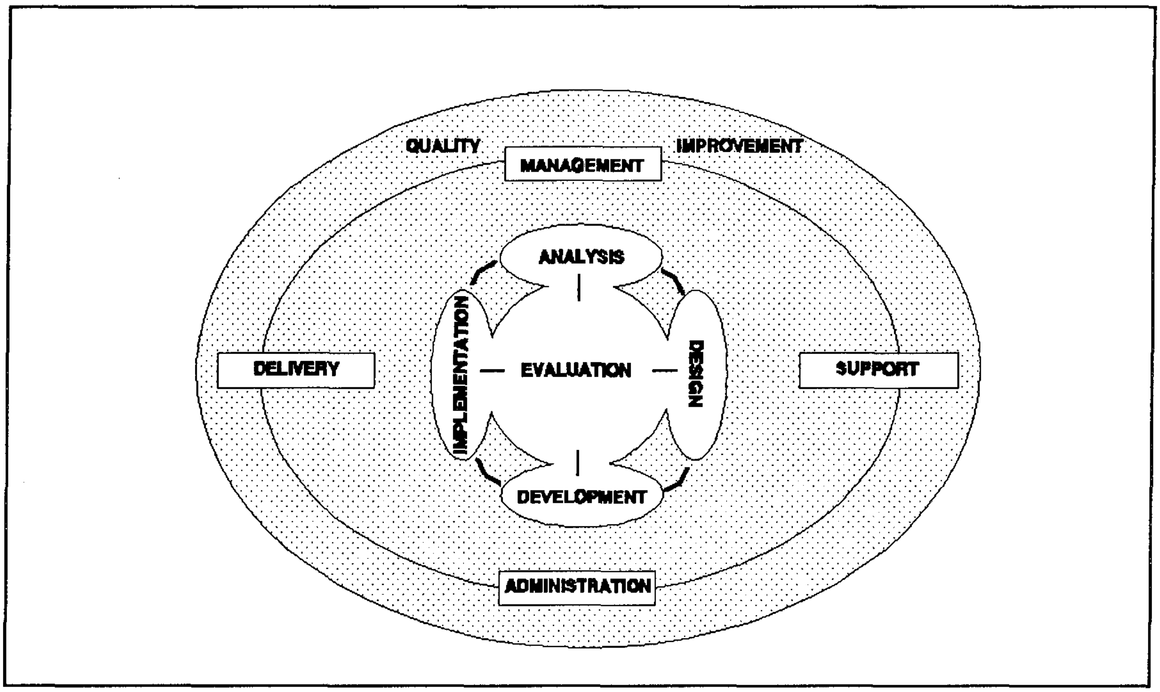


Figure 1. Air Force ISD Model

Systems Engineering

The widening of perspectives in the update of the Air Force ISD process takes into account the interactions between the ISD process and the systems engineering process in total training system acquisition (AF HANDBOOK 36-2235, Vol 3). The interactions shown in Figure 2 are an indication of the interchange of information at various phases of the development cycle. These interactions have functional dependencies which are managed as part of the core process integration in integrated product teams.

Common Reviews

The integration of processes also implements common reviews. The system requirements review at the beginning of the project brings all parties to an understanding of the end goal and the expect outcomes. The training system requirements analysis product reviews define the overall instructional system requirements and the preliminary instruction syllabus. The system design review addresses the specified approach for meeting the total instructional system functional requirements as well as the student training requirements. The preliminary and critical design reviews address the allocation of the system functional requirements or training requirements to each component of the instructional system, whether the component is hardware, software, or courseware, and the specified approach for meeting the component requirements. The review of the implementation plan covers the overall system integration and the interfaces for all components in the fully operational instructional system. The review of the test

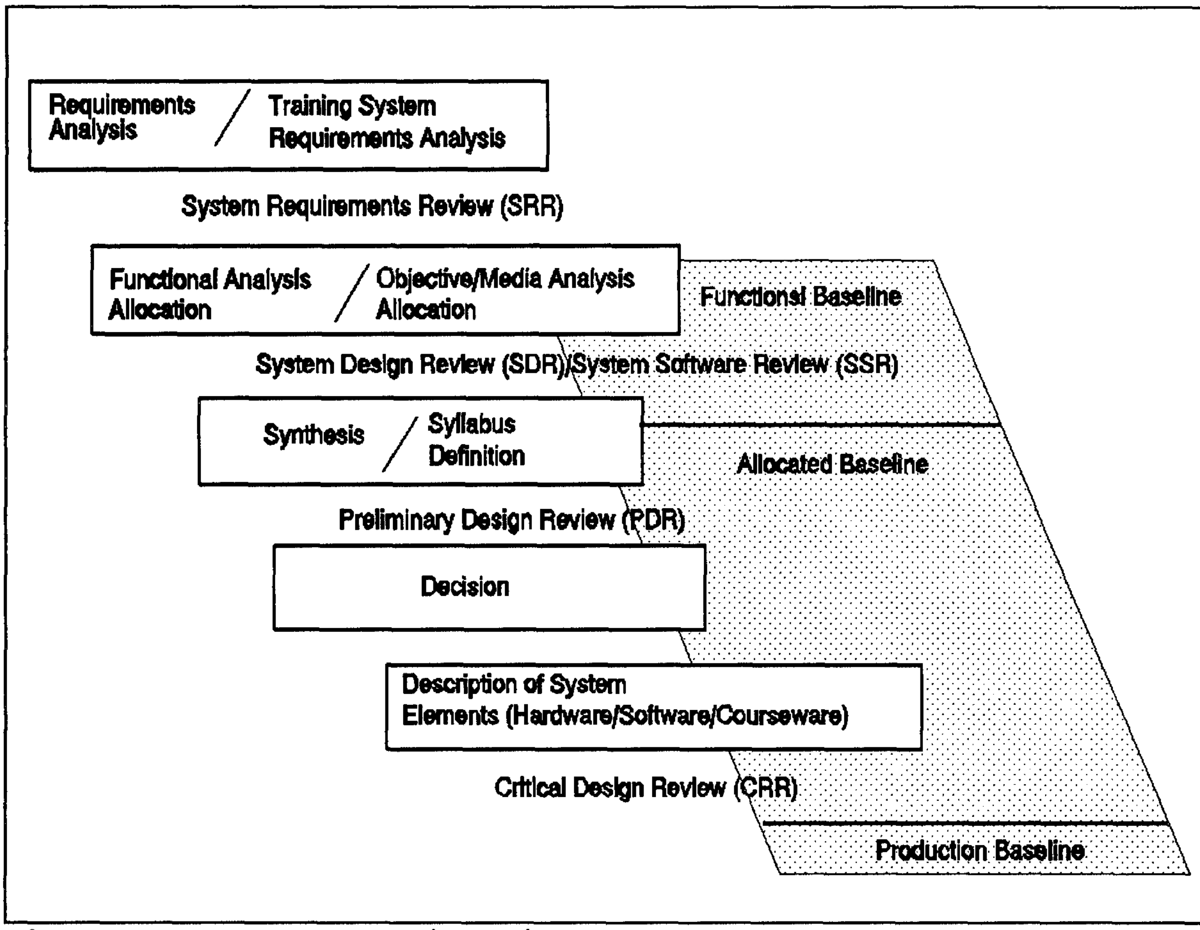


Figure 2. Systems Engineering/ISD Interchange

and evaluation plan ensures that the appropriate method is in place for assessing the achievement of the functional and operational requirements, at the system level and also at the component level (hardware, software, and courseware).

Requirements Analysis

The implementation of the expanded perspective guides the application of the ISD process. The training system requirements analysis phase leads first to an overall instructional system level or functional requirements definition and then to the allocation of system functions to the follow-on component level or instructional content requirements definition. The starting point of the system level definition ensures that the "big picture" encompasses the entire set of activities necessary for successful implementation, including the personnel assignments, facility needs, and life-cycle support requirements. The total system level perspective ensures the instructional system meets the functional requirements. The component level perspective ensures that the students, who are the product of the instructional system, do meet the training requirements. The overall effect is the implemented system meeting the operational requirment.

ANALYSIS COMPARISON

The expanded system level perspective for the ISD process also created a need for expanded analysis. Again borrowing from the systems engineering process, the approach adopted was similar to the engineering functional flow analysis. The functional analysis is a process for analyzing the operational requirements in terms of functions. This functional analysis consists of identifying functions, identifying the start and end conditions for each function, brainstorming more detailed activities or subfunctions that make up each function, and defining the sequence and relationships of the activities or subfunctions in a functional flow block diagram. Applied to the ISD process, the operational requirements for an instructional system are analyzed in terms of the identified functions for successful training systems. The analysis includes a review of the personnel organization planned to implement the operational requirements. The functional requirements defined through functional analysis drive the design of the overall instructional system architecture for the system level specification. The instructional system functions are allocated to the components, or the people who carry them out, and are then detailed in the specifications.

Functional Analysis

Functional analysis is different from the traditional mission/task analysis. Functional analysis is at the total training system perspective. In contrast, the mission/task analysis perspective is of the end job or activity to be performed as a result of instruction. The comparison of the target population analysis with the mission/task analysis leads to the definition of the training requirements. The training or instructional requirements drive the design of the curriculum. These requirements are allocated to the instructional products and then detailed in the instructional strategy.

A comparison of functional analysis with mission/task analysis is shown in Table 1. Although the actual data collected is different for each analysis, the data collection methods are similar. Functional analysis begins with the breakout of activities within each instructional system function (management, support, administration, delivery, evaluation). Also included are activities in the phases of the ISD process (analysis, design, development, implementation). Mission/task analysis begins with the breakout of the tasks/subtasks within a given mission or goal. Functional analysis maps the activities by function for the entire instructional system. Mission/task analysis maps the mission/task for the entire job or expected performance of graduates.

Process. The process for defining the functional requirements from a functional analysis is incorporated in the Air Force Aircrew Training System Guidespec. This process has been applied in the top-level design for the KC-135

Table 1
Analysis Comparison

Functional Analysis	**Mission/Task Analysis**
<u>What</u> List functions/activities Identify who performs Identify organization Identify processes Identify instructional system concept	<u>What</u> List mission/task Identify who performs Identify conditions Identify standards Identify knowledge and attitudes needed
<u>Getting started</u> Collect data Brainstorming with experts Observations Questionnaires Comparison-similar systems Review publications Review personnel orders Review organization Identify activities Hierarchy by function Validate Prioritize	<u>Getting started</u> Collect data Interview experts Observations Questionnaires Comparison-similar systems Review publications Review personnel orders Review organization Identify duties/tasks Hierarchy by duty Validate Prioritize
<u>Analysis</u> Activities within function Sequence of performance Concept of training comparison Process analysis Overall description	<u>Analysis</u> Component steps of tasks Sequence of performance Conditions for doing task Standards expected Knowledge/attitudes needed

Aircrew Training System (ATS), in the review 0and analysis of Educational Technology and Innovation at the United States Air Force Academy, and in the top-level study of United States Air Force Space Command organization for training systems.

Method. One approach detailing activities or subfunctions within each training system activity is to bring together key players and current documentation for the brainstorming session. The session begins by identifying any current instructional system activities. Documentation is referred to as needed to facilitate progress. Each function is written on a sticky-backed piece of paper and placed on a wall or large surface where it can be viewed by all present. The top level includes the functions of the ISD phases (analysis, design, development, implementation) and the basic instructional system functions (management, support, administration, delivery, evaluation). As each instructional system activity or subfunction is identified, the activity or subfuction is written on a sticky-backed piece of paper and placed under the most appropriate function. This process continues until those present feel the list is comprehensive.

Concept of Training. The idea of a training system concept was better defined during total training system acquisition. This concept needs to be derived in advance and all concerned should already be in agreement. The concept of training sets the bounds within which decisions can be made. Similar concepts from predecessor systems are already taken into account and projected instructional system high-drivers have been identified. A time continuum is included of the expected life-cycle of instruction a person is expected to follow. Basic instructional principles are listed which need to be emphasized and expected instructional outcomes and desired checkpoints have been identified.

Flow Diagrams. From the list derived from brainstorming, the current activities or subfunctions are compared to the concept of training, a comparison of actual to optimal. The list is resorted and adjusted, working toward an optimal solution. The activities or subfunctions are grouped by "process" in flow diagrams. Interfaces between activities or functions, the input/output, key decision points, and overall relationships with other functions are established. For example, the diagram could show the development function receiving direction to commence from the management function, confirming the training need, and then giving the results back to the management function. The Integration Definition for Information Modeling (1993) is a technique for function diagrams.

Information Mapping. The revised Air Force ISD process was published using an effective approach for documentation called information mapping (Horn, 1992). Information mapping sorts information into key blocks and provides a method for presentation of the information that is readily available to the user and quickly communicates the intent. Diagrams set up using this approach facilitate communication of analysis results and provide a baseline for development. As the instructional system design unfolds, these descriptions can then be expanded to support the implementation plan.

Training System Specifications

The training system guide specification (AFGS-87265, 1992) is organized to fit the expanded perspective required of total training system acquisition. The requirements section begins at the system level. Here the training system is defined as well as the interfaces of the training system with the air vehicle, support system, facility, and environment. System level characteristics are listed in context of the concept of training. Each of the training system functions is addressed. Following the overall system level requirements, the training requirements are described. These training requirements are used to derive performance requirements for lower level system/subsystems and prime item development specifications (PIDS). Training delivery components are described along with the training management and evaluation system and the training support system.

Test and Evaluation. The verification section is to assure that all training system requirements in the specification are met. Verification activities include both evaluation of the training system and evaluation of trainee performance. Verification activities include process and product evaluation, including interim discrete milestone evaluation points. All verification requirements at all specification levels are required to be traceable to the functional training system requirements. The quality assurance activity should be integrated with the configuration management activity such that any required functional configuration audit can serve both purposes. Each piece of the training system must successfully complete its test before higher level tests are conducted. This includes the hardware, software, and courseware, both individual and at various stages of integration. A full-scale evaluation of the total system in the fielded environment for an actual training period is to confirm compliance with the training system requirements. Since the requirements are mapped from the overall training system architecture, test and evaluation is also to assure that the training system functions are operational as designed.

Implementation Plan. The implementation plan guides the initiation of the designed training system functions. This plan can extend the function flow diagrams into subactivities needed to accomplish the overall system functions. This plan should unfold the "big picture" so that the integration of all training system components is carried out effectively and efficiently in day-to-day operations.

IMPACT SUMMARY

The step up to total training system acquisition did cause a significant learning curve in the application of the traditional ISD process. Advances made as a result of lessons learned are now incorporated in the updated Air Force ISD process and the application volumes that implement the process. The volume on application to acquisitions incorporates additional methodology from systems engineering and more closely aligns the ISD process with the systems engineering process. The guidance in these volumes has been field tested and is now available for future total training system acquisition programs.

REFERENCES

Bills, C. G., & Butterbrodt, V. L. (1992). Total trainingsystems design to function: A total quality management application. Proceedings of the National Aerospace Electronics Conference (NAECON), Dayton, OH, 1018-1023.

Dick, W. and Carey, L. (1990). Analyzing Jobs and Tasks.

Englewood Cliffs, New Jersey: Educational Technology Publications.

Fishburn, R. P., Williams, K. R., Chatt, J. A., & Spears, W. D. (1987). (AFHRL-TR-86-44). Design Specification Development for the C-130 Model Aircrew Training System: Phase I Report, Williams AFB, AZ: Air Force Human Resources Laboratory.

Gagne, R. M., Briggs, L. J., and Wager, W. W. (1992). Principles of Instructional Design (4th Ed.). New York: Harcourt Brace Jovanovich College Publishers.

Horn, R. E. (1992). Participants Manual for Developing Procedures, Policies and Documentation. Waltham, MA: Information Mapping, Inc.

Information for Designers of Instructional Systems: Application to Acquisition. (1993). (AF HANDBOOK 36-2235, Vol. 3). Washington, DC: Department of the Air Force.

Instructional System Development. (1993). (AFMAN 36-2234). Washington, DC: Department of the Air Force.

Integration Definition for Information Modeling (IDEF1X). (1993). (FIPS PUB 184). Gaithersburg, MD: Department of Commerce.

Merrill, M. D., Lee, Z., and Jones, M. K. (1990) Second Generation Instructional Design (ID2). Englewood Cliffs, New Jersey: Educational Technology Publications.

Reigeluth, C. M. (1983). Instructional design: What is it and why is it? In C. M. Reigeluth (Ed.), Instructional Design Theories and Models: An Overview of Their Current Status. Hillsdale, New Jersey: Erlbaum Associates.

Tennyson, R. D. and Michaels, M. (1991). Foundations of Educational Technology: Past, Present and Future. Englewood Cliffs, New Jersey: Education Technology Publications.

U.S. Air Force Guide Specification. (1992). (AFGS-87265). Wright-Patterson AFB, OH: Aeronautical Systems Center.

Williams, K. R., Judd, W. A., Degen, T. E., Haskell, B. C., & Schutt, S. L. (1987).

Advanced Aircrew Training System(AATS): Functional Design Description (TD 87-12). Irving, TX: Seville Training Systems.

AN ANALYSIS OF DISTANCE LEARNING APPLICATIONS FOR JOINT TRAINING

Kenneth P. Pisel, CDR, U. S. Navy
Armed Forces Staff College
Norfolk, Virginia 23511-1702

ABSTRACT

Today's constraints on manpower and funding have done little to constrain the ever-increasing demands for training. If we are to continue to meet these demands, innovation and technology must be applied through distance learning techniques to do more with less. Achieving the full potential of distance learning requires an analytical approach to selecting and implementing distance learning media. We must first understand the needs of the training program and the customers, media capabilities, and the costs and benefits of distance learning. But equally important is the knowledge of existing distance learning systems--how we can minimize costs and increase impact through interoperability. This paper models the analytical decision-making process used by the Armed Forces Staff College in evaluating distance learning alternatives to its current three-day traveling Joint Planning Orientation Course.

ABOUT THE AUTHOR

CDR Pisel has a BS in history from the United States Naval Academy, an MA in Personnel Management from Central Michigan University, and is a doctoral student at Old Dominion University. In 20 years in the Navy, he has operated as a helicopter pilot, logistics planner, and Naval Academy faculty member. He is currently a master faculty in the Department of Curriculum Development at the Armed Forces Staff College.

AN ANALYSIS OF DISTANCE LEARNING APPLICATIONS FOR JOINT TRAINING

Kenneth P. Pisel, CDR, U. S. Navy
Armed Forces Staff College
Norfolk, Virginia 23511-1702

In today's geopolitical-economic environment, effective implementation of the National Military Strategy depends on the synergistic efforts of the four Services. This synergistic approach to planning and executing military operations, referred to as "jointness" within the Department of Defense (DOD) and Congress, was mandated by the Goldwater-Nichols Department of Defense Reorganization Act of 1986. Among its provisions, the Goldwater-Nichols Act created a new occupational field for military officers: the Joint Specialty Officer (JSO). The JSO program is the foundation of effective joint planning and execution by major military staffs. However, because of output limitations in the Joint Professional Military Education system, only about 1,000 JSOs are produced per year.

To mitigate this shortfall, the Chairman of the Joint Chiefs of Staff directed the Armed Forces Staff College (AFSC) to design a short exportable course to provide orientation on joint planning. The Joint Planning Orientation Course (JPOC) was created to meet this requirement. The JPOC comprises twenty hours of orientation on the procedures used to create contingency plans in peacetime and operation orders in crises. Using two faculty members in a lecture format, the JPOC makes approximately thirty AFSC-funded and sixty user-funded (primarily Reserve Component units) trips per year. But even with these trips, only a fraction of the personnel that are either directly involved in or affected by joint planning are able to receive this training.

The potential demand generated by nine major unified commands containing over forty subordinate commands, with a vast array of Reserve Component support, is significant. A prime example of the JPOC's potential customer audience is the U.S. Army Reserve, which has personnel in over 7,000 units at 4,000 sites (Phelps, 1993). It is obvious that AFSC is unable to have a significant impact on a constituency of this magnitude with the current format of the JPOC. Constrained by manning and funding limitations, AFSC must look to innovation and technology to meet the joint training demands of the future. That promise cannot be fully realized, however, without an analytical approach to applying this technology.

In pursuing a technological answer to the challenges of education and training, four essential elements must be analyzed before recommending adoption of a specific distance learning system--program needs, media capabilities, systems interoperability, and costs. Needs analysis must address both the JPOC training program and the JPOC customer. Specific technical requirements for the program curriculum and scheduling must be compared with learning and scheduling needs of the students. Media analysis must include a review of the myriad options available, comparing their capabilities with program needs to identify the media most suitable for the program. Systems interoperability involves a review of distance learning systems currently in operation. Today all four military Services, industry, and educational institutions at all levels are engaged in distance learning programs--therefore systems interoperability will only serve to strengthen the field. Finally, a cost-benefit analysis is necessary to establish the viability of pursuing distance learning. This four-step process is described below in the analysis of distance learning for the JPOC program.

NEEDS

JPOC Curriculum

Requirements to support the JPOC curriculum are rather simple. The JPOC is a three-day, twenty-hour course designed to introduce the joint planning process to officer, enlisted, and civilian personnel. Course content is focused at the knowledge level of Bloom's taxonomy and is flexible only to the extent that the faculty member tailors it to the particular audience. As was noted earlier, presentation is via lecture by a traveling two-faculty-member team using approximately 700 slides. There are no exercises, simulations, or examinations in the course.

Customer Needs

Three groups of customers are served by the JPOC: personnel either assigned to or supporting major unified commands, personnel in the military's Reserve component, and foreign national military personnel. The primary focus of this paper is the Reserve Component. Although AFSC does not actively pursue Reserve participation, the JPOC focus on deployment planning is of great importance to Reserve units. The result is that over 60 percent of the JPOC presentations are to Reserve personnel. Current funding supports thirty annual trips to the major commands--both overseas and within the continental United States. Reserve unit access to the JPOC requires that those units furnish travel and per diem funding for AFSC instructors.

Needs for this audience tend to focus more on geography than on personal diversity. The JPOC military audience is adult, task-oriented, and fulfilling a training requirement. To meet these needs, a distance learning variation of the JPOC must have the following three capabilities: it must be capable of being presented over multiple time zones; it must be presented seven days per week, weekdays for the active personnel and weekends for reservists; and it must have a feedback capability to offset the ambiguity between doctrinally set procedures taught in the course and local procedures and nuances that vary worldwide.

MEDIA

"The best current evidence is that media are mere vehicles that deliver instruction but do not influence student achievements any more than a truck that delivers our groceries causes changes in nutrition."(Schlosser, 1994)

The grocery truck-media analogy illustrates the point that the distance learning medium is important only to the extent that it carries the desired message. However, it also makes the more subtle point that prospective students could suffer from intellectual malnutrition if the media-grocery truck never reaches them. To ensure that the customers receive the training product, decision-makers must understand what each medium has to offer (strengths and weaknesses), the audience it can reach, and how it can be applied to program requirements. Additionally, those making decisions must avoid the temptation to search for the one "best" medium to provide distance learning and then try to fit program requirements into that ideal.

Media Capabilities

Fourteen distance learning media are reviewed in this paper. Strengths and weaknesses of each medium are addressed in the context of the JPOC application (see Figure 1).

Comparative Analysis

Numerous media selection models compare program requirements with media capabilities to identify those media most compatible with a program. For the JPOC program a weighted model was used to make this comparison (see Figure 2 at Tab A). In this model preassigned values are listed under each medium for twenty-four individual design characteristics. These values range from zero to three, with zero assigned to those media that do not support a given design characteristic, increasing up to three as the media compatibility increases. Model weighting is accomplished through an "importance factor" selected by those knowledgeable about program requirements. In this instance, JPOC-qualified faculty were surveyed to determine importance factors for each design characteristic. The factor values, ranging from zero to ten, are multiplied by each preassigned value to create a weighted value for each design characteristic and medium.

The example below addresses the ability of each medium to be retained as a reference tool. Predetermined values range from zero to three. In this example, the non-resident seminar was valued at zero, indicating that it is not a good medium for retaining as a reference. Conversely,

Instructional Design Characteristics How important is it that the medium selected be able to:	0-10 Importance Factor	Print	Non-resident Seminar	Audio-graphics	Tele-course (2x2)
19. Be retained as a reference tool for later		3	0	1	2

print received a value of three because it is ideally suited as a tool for future reference.

Adapted from multiple sources

Medium	Strengths	Weaknesses
Print	. easy to update . low cost . allows self-pacing . adapts to student schedules . high-quality color graphics . can be retained as a reference . supports wide geographic dispersion	. lacks real-time faculty interaction . no instructional feedback . does not support remedial learning
Non-resident Seminar	. real-time faculty interaction . instructional feedback . responsive to curriculum changes . supports wide geographic dispersion	. not adaptable to individual learner needs . does not adapt to student schedules . cannot be retained as a reference . limited audience
Audio Tape	. easy to update . low cost . allows self-pacing . adapts to student schedules . responsive to curriculum changes . high-fidelity sound . can be retained as a reference . supports wide geographic dispersion	. lacks real-time faculty interaction . no instructional feedback . not adaptable to individual learner needs . no graphics
Videotape	. allows self-pacing . adapts to student schedules . high-quality color graphics . full-motion video . high-fidelity sound . can be retained as a reference . supports wide geographic dispersion	. lacks real-time faculty interaction . no instructional feedback . not adaptable to individual learner needs . requires production and editing capability
Hypertext	. allows self-pacing . adapts to individual learner needs . adapts to student schedules . high-quality color graphics . supports remedial learning . can be retained as a reference . provides rapid, nonlinear access to data . supports wide geographic dispersion	. lacks real-time faculty interaction . no instructional feedback . development costs . curriculum change costs . requires large amount of disk storage space
Tutorial CBI	. instructional feedback programmed . allows self-pacing . adapts to individual learner needs . adapts to student schedules . high-quality color graphics . supports remedial learning . can be retained as a reference . provides administration and off-line analysis of testing . supports wide geographic dispersion . provides real-time performance feedback	. lacks real-time faculty interaction . curriculum change costs . not designed for groups

Figure 1

Medium	Strengths	Weaknesses
Simulation/Gaming	. instructional feedback . allows self-pacing . adapts to individual learner needs . adapts to student schedules . high-quality color graphics . supports wide geographic dispersion . provides real-time performance feedback	. lacks real-time faculty interaction . does not support remedial learning . development costs . curriculum change costs . system maintenance cost
Intelligent CBI	. instructional feedback . allows self-pacing . adapts to individual learner needs . adapts to student schedules . high-quality color graphics . supports remedial learning . can be retained as a reference . provides administration and off-line analysis of testing . supports wide geographic dispersion . provides real-time performance feedback	. lacks real-time faculty interaction . development costs . curriculum change costs . not designed for groups
Multimedia/Hypermedia	. allows self-pacing . adapts to individual learner needs . adapts to student schedules . high-quality color graphics . full-motion video . high-fidelity sound . supports remedial learning . can be retained as a reference . provides rapid, nonlinear access to data . provides real-time performance feedback	. lacks real-time faculty interaction . curriculum change costs . dispersion limited by hardware requirements . development costs . equipment costs . requires large amount of disk storage space
Audio Conferencing	. real-time faculty interaction . responsive to curriculum changes . high-fidelity sound . supports wide geographic dispersion	. not adaptable to individual learner needs . does not adapt to student schedules . curriculum changes require major costs . no graphics
Audio-graphics	. real-time faculty interaction . responsive to curriculum changes . high-quality color graphics . full-motion video . high-fidelity sound . supports wide geographic dispersion	. does not adapt to student schedules . does not support remedial learning . cannot be retained as a reference . not designed for groups
Telecourse -- two-way video and two-way audio (2x2)	. real-time faculty interaction . responsive to curriculum changes . high-quality color graphics . full-motion video . high-fidelity sound . supports wide geographic dispersion	. not adaptable to individual learner needs . does not adapt to student schedules . requires specialized receiving equipment . equipment costs (higher than 1x2)

Figure 1

Medium	Strengths	Weaknesses
Telecourse -- one-way video and two-way audio (1x2)	. real-time faculty interaction . responsive to curriculum changes . high-quality color graphics . full-motion video . high-fidelity sound . supports wide geographic dispersion	. not adaptable to individual learner needs . does not adapt to student schedules . requires specialized receiving equipment . equipment costs
Computer-mediated Conferencing	. asynchronous feedback . responsive to curriculum changes . high-quality color graphics . full-motion video . high-fidelity sound . supports wide geographic dispersion	. does not adapt to student schedules . not designed for groups

Figure 1 (Air University, 1994, Bossinger and Milheim, 1993, and Zhang, 1994)

The importance factor is a subjective value assigned to each instructional design characteristic. This value is a judgment call and is set by those with knowledge and experience in the program. In the case of the JPOC program, a faculty survey set the importance factor for this characteristic at five.

As shown below in the second half of the example, the predetermined values for each medium are multiplied by the importance factor to reflect their true value to that program.

Instructional Design Characteristics How important is it that the medium selected be able to:	0-10 Importance Factor	Print	Non-resident Seminar	Audio-graphics	Telecourse (2x2)
19. Be retained as a reference tool for later	5	3 15	0 0	1 5	2 10

Weighted values were determined for the twenty-four instructional design characteristics in the media-selection model. The model identified telecourses, multimedia applications, computer-based instruction, and simulation/wargaming as the media most capable of supporting the JPOC requirements. These media and a combination of individually less compatible media will be reviewed for cost.

Interoperability

Taking advantage of existing systems is the educational equivalent of using a combat force-multiplier. This approach holds true regardless of the program, but it is even more important when trying to reach a multi-Service (joint) audience. The greatest common cost factor in any of the selected media is the start-up cost, and, when the program has to reach the joint community, worldwide interoperability becomes essential. With this in mind the plan was to look for off-the-shelf technology to minimize initial outlays and maximize impact. Discussion with Army, Air Force, and Navy training personnel identified outstanding alternatives readily available to AFSC. For a telecourse there are two sites willing to support AFSC within an hour of the college. To the north, the Army Training Support Center (ATSC) at Ft. Eustis has access to two separate networks: the Teletraining Network (TNET), with two-way video and audio, and the Satellite Education Network (SEN), with one-way video and two-way audio. To the southeast, the Navy's Chief of Naval Education and Training (CNET) operates a two-way video and audio system--the CNET Electronic Schoolhouse Network (CESN)--from Dam Neck in Virginia Beach. A fourth network already available at AFSC is the Defense Simulation Internet (DSI), but limited site access and operating costs make DSI less suited for JPOC application.

Operating compatible systems, TNET and CESN are leading the field in interoperability. TNET currently operates with SEN, DOD Video Teleconferencing, Iowa Information Network, Vermont Interactive Television, and

Kentucky Educational television, and it has plans to provide access to CESN for a total of almost 350 sites (Schall, 1994). TNET alone reaches over 110 Air Force Reserve and Army active duty, reserve, and National Guard sites. CESN currently reaches sixteen Navy sites, with expansion plans to access other naval sites worldwide (Ellis, 1994).

Ultimately, interoperability is an issue of cost and impact. By using the facilities of an existing system, and avoiding purchase or leasing costs, the JPOC program will save over $3.5 million (Ellis, 1994). The impact of interoperability brings to mind the earlier analogy of the delivery truck and the cliche about not reinventing the wheel. The option of creating a stand-alone system or interoperating with an existing system is comparable to making deliveries with a unicycle or with an eighteen-wheeler. It can be done either way, but it has a far greater impact using the latter.

Cost-benefit Factors

The final step in the analysis is to determine if the potential benefits accrued by distance learning outweigh the potential costs. A concept that cannot be funded remains only that--a concept. To this end, costs and benefits of the most compatible media are analyzed and compared.

Telecourse The greatest benefit of the telecourse is its ability to simultaneously reach multiple groups at remote sites, thereby eliminating travel and per diem costs for the faculty and at least minimizing these costs for the students. However, there is an ongoing debate on whether two-way or one-way video is better. An analytical approach to selecting learning media, however, removes the need for this debate. The decisions are made by analyzing program needs and comparing costs. Is there a need to be able to see the students? Does the program require the full-motion video that SEN provides? The issue is not which is better or best, but rather which is more suited and cost-effective for program needs.

For the JPOC program, operational costs can be minimized if AFSC uses CESN facilities at Dam Neck with its projected interconnectivity to TNET to access Navy, Army, and Air Force sites. CESN has operating capacity available and is fully funded by Navy sources through FY 95 (Ellis, 1994). This also makes the debate regarding two-way or one-way a moot point--two way is the CESN standard.

Multimedia The advantage of multimedia is the ability to reach those students who cannot reach network class sites. However, start-up costs for special equipment make this option less attractive. A system used by the Navy's medical community for training costs approximately $5,000 per copy. Funding for block purchase of these computer systems is beyond the current or projected funding capability of AFSC and it is unlikely that such funding would be readily available from other sources.

Development of the software for a multimedia system is time consuming and expensive, but there is an affordable option available from the Army. In addition to SEN and TNET, ATSC at Ft. Eustis has the military contract for the development of computer-based instruction. Cost for development of a multimedia course would be approximately $100,000 (Hiemstra, 1994).

Computer-based instruction CBI has two distinct cost advantages--it can be operated on computers used throughout the active duty and reserve communities, and the ATSC contract can develop a course for approximately $50,000 (Hiemstra, 1994). The goal would be to start small and place the software on floppy disks and, as CD-ROM became more available in the reserve community, to move in that direction. If supported, funding for CBI could be budgeted for and available by next fiscal year. The Internet could also be used in concert with CBI to allow interaction in a computer-mediated conference.

Simulation/wargaming Based on AFSC experience with wargaming, the costs in terms of manpower for development and maintenance would exceed the benefits available from these media.

Videotape/audio conference An economical option to reach groups that cannot connect with telecourse sites is to combine a JPOC videotape and an audio teleconference. Using existing AFSC capabilities, the JPOC could be taped and distributed to multiple sites. At a set date and time the tapes could be simultaneously started at all sites. Breaks for questions would be incorporated into the tapes and be conducted via audio teleconference. Since the cost for such conferencing is only around $35

Cost/Impact Comparison (1)

	Start-up Costs	Operating Costs	Travel	Total Sites	Average Students Per Site	Average Total Students	Total Costs	Average Cost Per Student
Computer-based Instruction	courseware development $20,000	none	none	distribute 5,000 copies of software	10	50,000	$50,000	$1.00
Videotape/ Audio Conference	negligible	$200 per course session	none	120 (2 sessions per month/ 5 sites each)	30	3,600	$4,800	$1.33
Telecourse (2) (either 2x2 or 1x2)	faculty training (travel) $3000	none	none	120 (2 sessions per month/ 5 sites each)	30	3,600	$3000	$0.83
Current Non-resident	none	none	$313K	90	40	3,600	$313,000	$86.94
Multi-media	courseware development $40,000	$5,000 per site	none	55	10	550	$315,000	$572.73

Figure 3

Notes: 1. This table does not address costs for program update. Update costs will range from non-resident seminar (least expensive) to multi-media (most expensive), but the actual costs depend on the magnitude of the change.
2. Reflects CESN/TNET facilities provided at no-cost. When AFSC on-site capability is leased cost-per-student rises to $27.50 for the first year and stabilizes at $21.11 for subsequent years.

per hour, five or six question sessions over three days would cost under $200 total.

A variation of this option would allow independent review of the videotaped lessons and asynchronous feedback. By using FAX, telephone, or the Internet those groups or individuals who cannot fit into a fixed schedule can still receive the material and ask questions off line. Feedback would not be instantaneous, but would be far better then none at all.

Current non-resident seminar format Data collected for the current non-resident format is not designed to provide accurate cost figures. AFSC retains data only on own-funded trips, so costs from those trips funded by the customers (over 60 percent) must be estimated. Using the known AFSC outlay of approximately $55,000 per year as a basis, it is estimated that customer-funded trips cost approximately $105,000 annually. A random sample of five trips indicates that an average of 10 percent of the students (360) travel at an average cost of $425 per student, for a total travel cost of approximately $153,000. Based on these figures, the annual cost for the current non-resident seminar is conservatively estimated at $313,000. An additional factor this figure does not include is the intangible cost of faculty time lost from the classroom.

Faculty training costs Accepting the truism that televised lessons can take good teachers and make them look better and take poor teachers and make them look worse, it is essential that AFSC train the faculty for the medium. The telecourse and videotaping formats require faculty training to produce well-planned and executed lesson programs. The Army Training Support Center at Ft. Lee, Virginia, offers a two-week program that is no cost to DOD agencies ($600 for civilians). Cost would be limited to travel and per diem of approximately $500.

Comparative analysis In the final analysis, it is a question of cost per student. Clearly, distance learning applications have the potential to deliver a similar or improved product to the same or larger audience at a fraction of the cost (see Figure 3). CBI, videotape/audio conference, and telecourses all take advantage of economical arrangements available from within AFSC or other DOD agencies. The low cost per student of the telecourse is made possible by using existing facilities from CESN and Navy funding; however, it is expected that a permanent arrangement would entail some form of fair-share cost borne by AFSC in the future. Additionally, economies of scale afforded by multiple users of CBI software make that option an even more attractive alternative. While the data admittedly represents a very broad estimate, the potential that it reflects is accurate.

THE FUTURE

The initial steps of the majority of distance learning programs tend to simply mimic the existing program. This phenomenon may be due to a lack of imagination on the part of distance learning planners, but a more rational explanation involves organizational comfort. The probability of encountering internal resistance when proposing radical change is significant. A more prudent approach employs an evolutionary change process with long-range goals. But even with evolutionary change, there will be an organizational paradigm shift (Moore, 1993). The applications of distance learning and technology require courses that are designed or redesigned to fit the media. These changes will surely not stop at the JPOC, but will spill over into the rest of the curriculum.

Such an evolutionary process was recommended for AFSC. Using three phases, the plan would program the implementation of a distance learning program over the next three years.

Phase I -- Foundations

- Coordinating with the National Defense University and the Joint Staff, AFSC must advertise the existence of the JPOC in publications such as the *Joint Force Quarterly* or in Reserve Component publications and bulletins. The magnitude of the demand must be more accurately measured.
- Begin faculty training. A nucleus of six instructors should be sent to the Army's two-week training course at Ft. Lee, Virginia.
- Modify graphics to support taped or televised format.
- Budget for developing a CBI program for the JPOC.

Phase II -- Initial Implementation

- Implement a videotape/audio conference JPOC program. Mimic the current JPOC lesson formats to increase the current audience without a significant change in procedures at AFSC.
- Develop a CBI curriculum for the JPOC.
- Implement a teleconference using the Navy's CESN at Dam Neck.

Phase III -- Long Range

- Implement a CBI JPOC.
- Implement this program at AFSC and report on the results in two years.

CONCLUSIONS

Technology is moving too quickly for organizations to jump into the distance learning arena and purchase equipment that could be obsolete within five years. It is more prudent and significantly more cost-effective to operate initially with a host facility; later, with success and growth of the program, equipment could be leased. If no host facility is available, the best alternative is to lease equipment interoperable with existing systems. AFSC has the luxury of excellent host facilities within easy reach that will allow the college to minimize cost and maximize impact. Ultimately, as AFSC and the rest of DOD plan for the future, we must strive to accomplish more with less--technology and innovation provide the means to do it.

As we move toward this goal it is wise to remember the words of B. H. Liddell Hart, who said "The only thing harder than getting a new idea into the military mind is to get an old one out." We must do both.

REFERENCES

Barker, Bruce O. Distance Education Technologies: All That Glitters Is Not Gold. ERIC, 1989. ED 309 894.

Barker, Bruce O., and Michael W. Dickenson. "Aspects of Successful Practice for Working With College Faculty in Distance Learning Programs." ED Journal Feb. 1994: 6-19.

Blevins, Larry. "Cost Benefit Analysis for Distance Learning." International Distance Learning Conference. Washington, D.C., 24 March 1994.

---. "Market Survey Analysis for Distance Learning." International Distance Learning Conference. Washington, D.C., 25 March 1994.

Bossinger, June, and William D. Milheim. "The Development and Application of Expert Systems." Educational Technology July 1993:7-17.

Chander, N. Jose. Management of Distance Education. New York: Sterling Publishers, 1991.

Chang, T.M., et al. Distance Learning. Boston: Kluwer-Nijhoff Publishing, 1983.

Charlton, James, ed. The Military Quotation Book. New York: St. Martin's Press, 1990.

Distance Learning Curriculum Analysis and Media Selection. Maxwell AFB: Air University, 1994.

Dumestre, Jeanie. "Two-way Interactive Video Teletraining." Proceedings: Conference on Distance Learning in DOD. Washington, D.C.: National Defense University, 1993.

Dunn, Karen. "TRADOC Video Teletraining (VTT)." Proceedings: Conference on Distance Learning in DOD. Washington, D.C.: National Defense University, 1993.

Ellis, Jean C., CESN Project Manager. Personal interview, 31 May 1994.

Flynn, Lin. "Defense Simulation Internet." International Distance Learning Conference. Washington, D.C., 23 March 1994.

Franchi, Jorge. "Virtual Reality: An Overview." Tech Trends Jan/Feb 1994: 23-26.

Golas, Katherine C. "Estimating Time to Develop Interactive Courseware in the 1990s." Proceedings: 15th I/ITSEEC Conference, 1993.

Hiemstra, Donald, ATSC, Acquisition Support Branch. Telephone interview, 18 August 1994.

Johnson, Kenneth. "Distance Learning, Educating the Television Teacher." Proceedings: Conference on Distance Learning in DOD. Washington, D.C.: National Defense University, 1993.

Kaye, Anthony, and Greville Rumble. Distance Teaching for Higher and Adult Education. London: The Open University Press, 1981.

McGreal, Rory. "Pedagogical Screen Design Principles for Graphics in Teleconferencing." ED Journal Feb. 1994: 10-13.

McNamarra, Michael R. "Teleconferencing in Support of Distance Education with the Army Forum." Proceedings: Conference on Distance Learning in DOD. Washington, D.C.: National Defense University, 1993.

Mills, Steven C. "Integrated Learning Systems." Tech Trends Jan/Feb 1994: 27-28.

Moore, Michael G. "Is Teaching Like Flying? A Total Systems View of Distance Education." The American Journal of Distance Education 7 (1993): 1-10.

Moore, Michael G., and G. Christopher Clark. Readings in Distance Learning and Instruction. University Park: American Center for the Study of Distance Education, 1989.

Ostendorf, Warren. "Interactive Compressed Video Distance Learning and the Adult Learner." Association for Educational Communications and Technology. Nashville, 16 Feb. 1994.

Phelps, Ruth. "Asynchronous Computer Conferencing for Reserve Component Training." Proceedings: Conference on Distance Learning in DOD. Washington, D.C.: National Defense University, 1993.

Romiszowski, Alexander. Telecommunications and Distance Education. ERIC, 1993. EDO-IR-93-2.

Schall, Keith, ATSC TNET Coordinator. Personal interview, 17 June 1994.

Schall, Larryetta, ATSC, Chief Acquisition Support Branch. Personal interview, 17 June 1994.

Schlosser, Charles A., and Mary L. Anderson. Distance Education: Review of the Literature. Ames: Association for Educational Communications and Technology, 1994.

Schrum, Lynne. Distance Education: A Primer for Administrators. ERIC, 1991. ED 336 866.

Shepard, Scott. Telephone interview. 31 Mar. 1994.

Studebaker, James. "Navy Health Science's Education and Training." International Distance Learning Conference. Washington, D.C., 23 Mar 1994

Wagner, Ellen D. "Variables Affecting Distance Educational Program Success." Educational Technology April 1993:28-32.

Willis, Barry. Distance Education: A Practical Guide. Englewood Cliffs: Educational Technology Publications, 1993.

Wilson, Chuck. Trends in Distance Education: A Viable Alternative for Higher Education. ERIC, 1991. ED 337 081.

Wolcott, Linda L. "Faculty Planning for Distance Teaching." The American Journal of Distance Education 7 (1993): 26-35.

Zhang, Shuqiang. "TV Instruction With Two-way and One-way Video." Association for Educational Communications and Technology Convention. Nashville, 17 Feb. 1994.

JPOC Media-selection Model *

Instructional Design Characteristics How important is it that the medium selected be able to:	**0-10 Importance Factor**	Print	Non-resident Seminar	Audio Tape	Video-tape	Hyper-text	Tutor-ial CBI	Simu-lation Gam-ing	Intelli-gent CBI	Multi-media/ Hyper-media	Audio Confer-ence	Audio-graphics	Tele-course 2x2	Tele-course 1x2	Compu-ter Med-iated Confer-ence
1. Provide real-time interaction between students and instructors	**8**	0 **0**	1 **8**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	1 **8**	0 **0**	3 **24**	3 **24**	3 **24**	3 **24**	3 **24**
2. Provide real-time interaction with other students (group interaction)	**6**	0 **0**	3 **18**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	2 **12**	2 **12**	3 **18**	3 **18**	2 **12**
3. Provide immediate instructional feedback from programmed or system (non-human) sources	**4**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	3 **12**	2 **8**	3 **12**	3 **12**	0 **0**	0 **0**	0 **0**	0 **0**	2 **8**
4. Allow self-pacing	**3**	3 **9**	0 **0**	3 **9**	3 **9**	3 **9**	3 **9**	3 **9**	3 **9**	3 **9**	1 **3**	0 **0**	0 **0**	0 **0**	0 **0**
5. Allow learner control of scope and sequence	**2**	0 **0**	0 **0**	0 **0**	0 **0**	3 **6**	2 **4**	3 **6**	1 **2**	2 **4**	1 **2**	1 **2**	1 **2**	1 **2**	2 **4**
6. Support individualized instruction (ability to adapt to learner needs)	**4**	0 **0**	1 **4**	0 **0**	0 **0**	2 **8**	2 **8**	3 **12**	3 **12**	3 **12**	1 **4**	1 **4**	1 **4**	1 **4**	1 **4**

Figure 2

*Adapted from Distance Learning Curriculum Analysis and Media Selection. Maxwell AFB: Air University, 1994.

Tab A

Instructional Design Characteristics How important is it that the medium selected be able to:	**0-10 Importance Factor**	Print	Non-resident Seminar	Audio Tape	Video-tape	Hyper-text	Tutor-ial CBI	Simu-lation Gam-ing	Intelli-gent CBI	Multi-media/ Hyper-media	Audio Confer-ence	Audio-graphics	Tele-course 2x2	Tele-course 1x2	Compu-ter Med-iated Confer-ence
7. Support privacy of student inquiry/ response	**2**	3 **6**	0 **0**	1 **2**	1 **2**	3 **6**	3 **6**	3 **6**	3 **6**	3 **6**	0 **0**	0 **0**	0 **0**	0 **0**	2 **4**
8. Fit various student schedules	**4**	3 **12**	2 **8**	3 **12**	3 **12**	3 **12**	3 **12**	3 **12**	3 **12**	3 **12**	2 **8**	1 **4**	1 **4**	1 **4**	1 **4**
9. Allow short-notice changes to the curriculum	**6**	1 **6**	2 **12**	2 **12**	1 **6**	2 **12**	2 **12**	1 **6**	1 **6**	1 **6**	3 **18**	2 **12**	3 **18**	3 **18**	2 **12**
10. Support inclusion of realism and simulation type scenarios	**6**	1 **6**	2 **12**	1 **6**	2 **12**	1 **6**	2 **12**	3 **18**	2 **12**	3 **18**	1 **6**	2 **12**	2 **12**	2 **12**	2 **12**
11. Support inclusion of high-quality color graphics	**8**	3 **24**	0 **0**	0 **0**	3 **24**	3 **24**	2 **16**	2 **16**	2 **16**	3 **24**	0 **0**	2 **16**	2 **16**	3 **24**	1 **8**
12. Support full-motion video	**5**	0 **0**	0 **0**	0 **0**	3 **15**	0 **0**	0 **0**	0 **0**	0 **0**	3 **15**	0 **0**	2 **10**	2 **10**	3 **15**	0 **0**
13. Support transfer of rapidly changing electronic data	**3**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	0 **0**	3 **9**	0 **0**	2 **6**	2 **6**
14. Provide high-fidelity sound	**7**	0 **0**	0 **0**	3 **21**	3 **21**	0 **0**	1 **7**	1 **7**	1 **7**	3 **21**	3 **21**	3 **21**	3 **21**	3 **21**	3 **21**

Figure 2

Tab A

Instructional Design Characteristics How important is it that the medium selected be able to:	**0-10 Impor-tance Factor**	Print	Non-resident Seminar	Audio Tape	Video-tape	Hyper-text	Tutor-ial CBI	Simu-lation Gam-ing	Intelli-gent CBI	Multi-media/ Hyper-media	Audio Confer-ence	Audio-graphics	Tele-course 2x2	Tele-course 1x2	Compu-ter Med-iated Confer-ence
15. Support remedial learning or reteaching	**4**	1 **4**	2 **8**	1 **4**	1 **4**	2 **8**	3 **12**	2 **8**	3 **12**	2 **8**	2 **8**	2 **8**	2 **8**	2 **8**	2 **8**
16. Support higher levels of cognitive learning (application level or higher in Bloom's taxonomy)	**3**	2 **6**	3 **9**	2 **6**	2 **6**	1 **3**	3 **9**	3 **9**	3 **9**	3 **9**	3 **9**	3 **9**	3 **9**	3 **9**	3 **9**
17. Support affective domain learning objectives	**5**	2 **10**	3 **15**	1 **5**	2 **10**	1 **5**	2 **10**	3 **15**	2 **10**	3 **15**	2 **10**	1 **5**	3 **15**	3 **15**	2 **10**
18. Support psychomotor domain learning objectives	**2**	1 **2**	1 **2**	1 **2**	2 **4**	1 **2**	2 **4**	3 **6**	2 **4**	3 **6**	1 **2**	1 **2**	3 **6**	2 **4**	1 **2**
19. Be retained as a reference tool for later	**5**	3 **15**	0 **0**	3 **15**	3 **15**	3 **15**	2 **10**	1 **5**	2 **0**	3 **15**	2 **10**	1 **5**	2 **10**	2 **10**	2 **10**
20. Promote rapid, non-linear access to information	**5**	1 **5**	2 **10**	0 **0**	0 **0**	3 **15**	1 **5**	1 **5**	1 **5**	3 **15**	2 **10**	2 **10**	2 **10**	2 **10**	2 **10**

Figure 2

Tab A

Instructional Design Characteristics How important is it that the medium selected be able to:	0-10 Importance Factor	Print	Non-resident Seminar	Audio Tape	Video-tape	Hyper-text	Tutorial CBI	Simulation Gaming	Intelligent CBI	Multi-media/ Hyper-media	Audio Conference	Audio-graphic	Tele-course 2x2	Tele-course 1x2	Computer Mediated Conference
21. Support testing administration, documentation of performance, off-line analysis	4	0 0	0 0	0 0	0 0	0 0	3 12	2 8	3 12	1 4	1 4	1 4	2 8	2 8	3 12
22. Support wide geographic dispersion	9	3 27	2 18	3 27	3 27	3 27	3 27	3 27	3 27	1 9	2 18	2 18	1 9	3 27	1 9
23. Support sporadic, uneven student loads	5	3 15	1 5	3 15	3 15	3 15	3 15	3 15	3 15	3 15	1 5	1 5	1 5	1 5	1 5
24. Provide real-time performance feedback to students	5	0 0	0 0	0 0	0 0	0 0	3 15	3 15	3 15	3 15	2 10	2 10	2 10	2 10	2 10
Totals		147	127	136	182	169	206	213	221	250	186	202	219	256	204
Compatibility ranking with JPOC		12	14	13	10	11	6	5	3	2	9	8	4	1	7

Figure 2

Tab A

PARTNERS IN EDUCATION
CHANGING THE WAY STUDENTS LEARN

Michael D. Williams
Naval Air Warfare Center Training Systems Division
Orlando, Florida

Marsha C. Vandivort
Edgewater High School
Chairman, Visual Arts Department

Jason Ahmanson
Edgewater High School
Student

ABSTRACT

The overwhelming ills faced by education today will never be cured by using outdated traditional processes for education. Worn-out lectures, tests and homework fall far short in challenging high school students to learn the skills they desperately need to face their rapidly changing future. The process is changing in Orange County, Florida, where the Naval Air Warfare Center Training Systems Division (TSD) and Edgewater High School, supported by Apple Computer, Inc., have joined under the *Partners in Education Agreement* to provide a new learning paradigm in one classroom environment.

The Training Systems Division needed a method to explain the underlying concepts of Distributed Interactive Simulation (DIS) and Edgewater High School was looking for ways to utilize their computer animation lab. Edgewater and NAWCTSD jointly planned a learning venture for the students to produce the DIS Instructional Animation. This project provided a "real-world" multi-media production which would enrich students' skills in visual arts, group dynamics, computer operation, and problem solving in a multi-disciplined team environment.

Students were encouraged to learn to structure a task from conception to completion, work in groups and independently, communicate ideas verbally and visually, manage time and set priorities. Students reported learning important skills from participation in this project such as: cooperation, drawing, color theory, organization, public speaking, advertising, brainstorming, animation, working with others, problem solving and business planning. What started as a simple classroom project, evolved in into a revolutionary teaching and learning experience.

ABOUT THE AUTHORS

Michael D. Williams is the DIS Lead Engineer in the Systems Simulation Branch at the Naval Air Warfare Center Training Systems Division. He has specialized in real-time simulation modeling and Distributed Interactive Simulation. Mr. Williams has a Masters Degree in Computer Engineering from the University of Central Florida in Orlando, Florida. His Bachelors Degree in Electrical Engineering is from Old Dominion University in Norfolk, Virginia. Email: williamsm@ntsc.navy.mil

Marsha C. Vandivort is a producing artist, fine arts administrator, and visual arts instructor. Her formal education includes painting studies in Germany, graduation from Northeast Missouri State University, post-graduate work at the University of Missouri, Southern Illinois Artists Workshop, Savannah College of Art and Design, and Internships with William Unger and Trevor Southey. She is currently Visual Arts Chairman and computer graphics instructor at Edgewater High School (Orlando, Florida) and a computer graphics curriculum/assessment writer for the Florida Educational Technology Quarterly.

Jason Ahmanson is currently a student at Edgewater High School and was one the student corporate presidents.

PARTNERS IN EDUCATION
CHANGING THE WAY STUDENTS LEARN

Michael D. Williams
Naval Air Warfare Center Training Systems Division
Orlando, Florida

Marsha C. Vandivort
Edgewater High School
Chairman, Visual Arts Department

Jason Ahmanson
Edgewater High School
Student

"*We are moving in a new direction to create an educational and training system that challenges American workers to match their skills to the demands of a fast-paced economy and challenges our students to reach for resources beyond their classrooms.*"

President William J. Clinton
***"Technology for America's Economic Growth",* Feb. 22, 1993**

INTRODUCTION

Naval Air Warfare Center Training Systems Division (formerly the Naval Training Systems Center) develops, acquires, and maintains training systems for the Navy and Marine Corps. The Training Systems Division, in Orlando, Florida is a leader in Department of Defense Distributed Interactive Simulation (DIS) research and development efforts and serves as the DIS Agent for the Navy. DIS is a new and unique method of connecting dissimilar simulators, strategic planning systems, and instrumented equipment over an electronic network for combined exercises.

Edgewater High School (EHS) has been a part of Florida's Orange County Public School system since 1952. In response to an awareness of the high caliber of students in its community and throughout Orange County, EHS created the Engineering Science and Technology (EST) Program in August, 1991. With the success of the EST program, Edgewater has continued to embrace opportunities to challenge its students using innovative programs and by developing educational partnerships with community organizations. Educational partnerships between public schools, students, parents, and business can provide experiences, including those outside the formal classroom, that empower student learning through mentorships, cross-disciplinary studies and collaborative projects centered around advanced technology topics.

Naval Air Warfare Center Training Systems Division (TSD) initially joined Edgewater High School when the *Partners in Education Agreement* was signed with the Orange County Public School System in December, 1991. The original agreement provided for speakers, demonstrations and field trips to the TSD. As an outgrowth of this partnership, the Distributed Interactive Simulation Instructional Animation Project was conceived in January, 1993. This project was designed to provide EHS students with an opportunity to participate in "real-world" multi-media production which would enrich their skills in visual arts, group dynamics, computer operation, and problem solving in a multi-disciplined team environment. To facilitate communication, Edgewater was provided with a computer account on the TSD research VAX computer for an electronic mail link with the TSD engineer and access to the Internet.

BENEFITS OF OUR EDUCATIONAL PARTNERSHIPS

This animation project was facilitated through the joint efforts of professionals from TSD, Orange County Public Schools, Edgewater High School, and Apple Computer, Inc®. Even though facilitators provided technical advice, it was an important objective of this project that students be responsible for all creative aspects as well as the technical production of the Instructional Animation.

The students were introduced to many technical concepts of DIS during a formal presentation at the TSD. This meeting allowed Michael Williams to explain his needs and meet the student participants. The students were provided with information regarding the mission of the TSD, the concepts of training and simulation, and the principles of DIS. A timeline was presented to the students including several incremental milestones and a delivery date for the final product. Following the presentation, students discussed the feasibility of an animation project and decided to commit their time and creative efforts to it. The meeting concluded with a student brainstorming session to conceptualize the animation.

The success of this project was due, in part, to the contributions and efforts of many people. Throughout the project's lifespan, the students were assisted by subject matter experts in the areas of engineering and DIS, computer technology, creative writing, theater, and visual arts. Parents volunteered clerical skills, transportation, food, technical skills, equipment, and family time. Their support was invaluable.

The *Partners in Education* program transformed a classroom learning environment at Edgewater High School into a "real world" workplace. This program also benefited the Training Systems Division by providing a method of explaining abstract aspects of networking and Distributed Interactive Simulation. What started as a simple classroom project, evolved in into a revolutionary teaching and learning experience.

DISTRIBUTED INTERACTIVE SIMULATION

The Department of Defense has invested many resources in a wide variety of simulators currently located throughout the world. This investment of technology has provided our military with exceptional training in realistic, but simulated, situations. However, as the different military agencies have begun to operate in combined efforts, the need for training in coordinated missions has become essential. The Department of Defense wanted a way to interconnect these existing simulators and allow them to interact with one another in the same way that an air wing of F-18s interacts with an aircraft carrier in a battle group. The Defense Advanced Research Projects Agency (DARPA) began investigating Simulation Networking (SIMNET) in the late 1980s. Out of the SIMNET effort grew Distributed Interactive Simulation in 1990.

The primary mission of Distributed Interactive Simulation (DIS) is to create synthetic, virtual representations of warfare environments by systematically connecting separate subcomponents of simulation which reside at distributed, multiple locations. Basically, this means that simulation systems located at many different remote sites can interact with one another as if they were located in the same building. F/A-18 pilots training on a flight simulator at NAS Oceana, Virginia can train with other F/A-18 pilots training on simulators in Mayport, Florida. Within the simulation, each pilot can see the other aircraft. These simulations can also be linked and coordinated with training being conducted at other locations such as NAS North Island, in San Diego or any where else.

Distributed Interactive Simulation spans several vital issues: (1) the simulation systems connected on the network, (2) Protocol Data Units for information transfer via the network, (3) the network used to connect simulation systems, (4) the virtual environment in which the simulation takes place. The simulation systems which are being connected on the network are used to generate aircraft, ships, tanks, and infantry. These objects are given the generic name <u>entities</u>.

Protocol Data Units (PDUs) are used to package the information sent between the simulation systems. Each PDU contains information identifying the sender, receiver, and other information based on the type of PDU. For example, the Entity State PDU, the most frequently used PDU, includes the sender's network address, type of entity (F-14, M1, etc), XYZ location, XYZ velocities, and other critical information.

Simulation systems are connected via computer networks. The most common type used today is the Ethernet Local Area Network because of its widespread use, low cost, reliability, and ease of use. The local area network (LAN) is used to connect computer systems which are located within close proximity to one another (in the same building). Other network architectures are used to allow simulation systems to communicate more information faster or over longer distances. Fiber optic cable is currently being used to connect computer systems which need to pass larger amounts of data faster. The Internet and the Defense Systems Internet (DSI) allow computers to communicate across the United States and farther.

In order to structure a training exercise, all entities must come together in the same area of the world. This area is referred to as the virtual environment. The virtual environment is a computer database modeled after a real place in the world. For the 1992 and 1993 Interservice/Industry Training Systems and Education Conferences, (I/ITSEC) we used a 100 square mile area on the coast of California called Fort Hunter Liggett. This database outlined the California shoreline, the terrain elevations, and terrain features such as roads, trees, and buildings existing on the real Fort Hunter Liggett area.

DIS has some key features which increase the useability, interoperability, and fidelity of simulation exercises: (1) There is no central computer controlling the simulation. Each entity must bring its own processing power or computer capability. (2) Each entity, or node, is autonomous and is responsible for maintaining the state of its entity(ies). (3) There is a standard protocol for communication with the DIS environment. (4) Each entity, or node, is responsible for determining what part of the simulation it perceives. (5) Dead reckoning is used to reduce the amount of network communication necessary.[2]

THE INSTRUCTIONAL ANIMATION PROJECT

For many years the Navy has relied on standard media to communicate ideas, projects, and designs. While video is being used more frequently, the standard presentation tools are overhead projectors, brochures, and documents. As the technology being communicated has grown increasingly more complex, the methods of communication have remained the same. Distributed Interactive Simulation is an example of one of those new technologies which usually requires hours of explanation for new users to visualize entities, simulators, and virtual battlefields.

Edgewater High School was exploring possibilities of computer animation via an educational grant from Apple Computer, Inc®. When Marsha Vandivort and Michael Williams sat down and began discussing the feasibility of communicating the complex ideas of DIS through animation, the two technologies seemed a perfect match.

The DIS Instructional Animation Project employed computer animated sequences and voice clips to explain the underlying concepts of DIS. The Edgewater student group created two characters, a blob and a janitor, who interact through a cartoon format to explain DIS Entities, the DIS communication protocol, and the different methods of networking. The animated cartoon depicts the blob leading the janitor on a journey around the DIS system while explaining three main areas of DIS. This journey takes the characters into the network cables, out to a real battlefield with simulated aircraft, and inside of a computer system.

This project provided a unique learning platform, giving the students insight into a current Department of Defense research effort, the operation of Apple computers, and the dynamics of multi-media. The primary objective of this project's founders was to make this entire production as true to the real world as possible.

Students were encouraged to work in groups as well as independently, learn to structure a task from conception to completion, communicate ideas verbally and visually, manage time, set priorities, and master computer skills. To provide the students with a non-traditional, highly effective learning environment, the organization of this student team was modeled after that of a real corporation. The corporate structure included the customer, employees, management and executive officers.

The customer was the TSD. The president of the corporation was the visual arts instructor, Marsha Vandivort. The students made up the employee population with four of them serving as vice presidents. Student executive responsibilities included communicating information, such as corporate and client meetings and delegating assignments from the president. Although all students were involved with the design process, certain student management teams provided Quality Control, Sound Editing, and Scripting.

The real-world business organization of work allowed students scheduled in different classes at different times of day to coordinate their work. Because they scheduled their own deadlines according to a production delivery date, their abilities grew rapidly in totally new learning areas. Virtually every skill needed for the entire project was a new one for the group.

The fact that the corporation had the capability to interview, hire, promote, or dismiss employees based on production needs and work performance, helped motivate all students to carry their share of the workload. Regular corporate meetings were held to make work assignments, prioritize time, delegate project responsibility, critique the animation, and evaluate progress. Student communication and organizational skills improved rapidly as they assumed their full corporate responsibilities.

STUDENT DEMOGRAPHICS

This student corporation provided a unique and diverse learning environment. While student ages ranged from 13 to 17 years old, most students were 15 or under at the start of the project. They were born in six different states and hailed from five countries. Their native languages included English, Hungarian, Popumentu, Vietnamese, French, Spanish and Farsi.

Although a few students were first introduced to the Macintosh computer as early as 1988, most had their first computer experience just a few months prior to starting work on this animation project. Only two members of the group had any experience with the software used to create the animation or with sound editing. In addition, most students had a minimum of art study and some had none at all. Because students' abilities varied throughout the group, students rapidly learned that they needed to share their knowledge with one another. This peer-level instruction maximized student expertise and became a valuable component of this learning process.

Emerging visual arts technology today provides an expansive opportunity for personalizing study to meet the specific needs of each student, from exceptional and learning disabled to gifted and talented. In this project, traditional teaching and learning methods were abandoned in favor of more efficient roles for both teacher and student. The teach was no longer the "expert", but rather became the "stage manager" in a theater where students became problem-solvers and developed skills for life-long learning.

In addition to the established partners in the project, the expertise of a wide variety of professional in the community was utilized: a children's author/illustrator taught storyline development, a film-maker showed students how to sequence video, a businessman calibrated the color printer and a local musician helped students learn to compose music on a computer.

Students said they learned the following important skills from participation in this project: cooperation, drawing, color theory, digital sound, organization, public speaking, advertising, brainstorming, animation, working with others, problem solving and business planning. They said developing imagination, flexibility, leadership, and responsibility were real world competencies which were gained.

The most dramatic changes came in the growth of students as individuals. The girl who started the project with red dreadlocks, blue combat boots and a nose ring developed enough self-esteem to buy a business dress and speak before the school board. A learning-disabled student became a specialist in mechanical illustration and was sought out by other students when they had drawing problems. And a student on probation for gang-related violence apologized when he had to leave group work to do community service. The proof, in this case, is more **in the process** than **in the product**. Combining the know-how of the professional business world and government with the strengths of traditional education and packaging it in a project-driven format can change both the scope of education and the futures of the students involved.

The students carry a normal academic workload, spend their spare time painting, running, skating, performing music as well as working on computers. However, they have devoted countless hours for the successful completion of this project They anticipate careers in engineering, military aviation, computer repair, fine arts, physical therapy, advertising, design, and animation. No matter what their vocational choices, clearly this experience will contribute to their future success.

Students say that they have "re-approached coursework by learning how to accomplish, instead of simply learning how to retain." According to one student, he will "know how to work in tough situations, about down time and short deadlines" and "to share what [he's] learned with others and to learn from them in order to get a job done." When students express the impact the project has had on their futures, they site "develop[ing] personality traits like patience, cooperation, self-confidence and flexibility" as important things the project provided.

By having input into what they learn and do each day, students maintain that they have mastered far more skills than they would have in a traditional classroom and have definitely worked harder <u>and</u> had more fun. In forecasting the corporation's future, they see their corporation as an ongoing concern, continuing to seek new clients, with new projects, offering new challenges as the years pass. They hope that their student successors will have the same opportunities to innovate and excel with future projects.

A STUDENT VIEW

When asked to describe this learning experience, Jason Ahmanson responded with the following:

"This project has helped me learn how a corporation works. We were divided into groups, each having a 'vice president'. I began as a worker, but then, the vice president of my group couldn't attend our summer class, so we decided to vote on a new vice president. We had one of our many corporate meetings and whomever wanted to be a vice president, including me, made a statement of our qualifications and then left the room. The remaining people took a vote and I was chosen to be the new vice president. This experience helped me to become a better leader rather than a follower.

I've since better learned how to work in a group along with increasing my communication skills, which I needed since a major part of this project was continuity. It was quite hard to keep the characters and background looking the same since we had many different artists, each with different skills and ideas, who didn't draw the same.

We had to solve many problems to accomplish this animation. For example, the greatest one was the technology we had to work with. While our computers were great, they couldn't stand up to this big project. We were constantly running out of memory and files were being lost for reasons still unknown.

Another problem, sound editing, just ate up our limited memory even more and forced us to employ another entire team. We had one of Macintosh's best computers, the Centrus 650, and we still filled it up. So it's safe to say that technology was one of our greatest barriers. However, with hard work and perseverance, we have overcome most of these obstacles.

We were not all artists when we first got involved. In fact, some of us didn't know the first thing about art. Then again, others didn't know the first thing about computers. We first learned a little about each other, and shared our ideas about the subject we knew best with the 'professionals' in the other areas. I think we all turned out being good artists, and computer technicians while learning to put all of our skills together toward a single goal.

In conclusion, I found this project to be hard at times, but then again, nothing's rewarding unless you work for it."

LOGISTICS

The DIS Instructional Animation project began during the spring semester of the 1992-93 school year. As the school administration began to see the importance and scale of this project's undertaking, a decision was made to introduce a three-week summer school program specifically geared to this project. The students met seven and a half hours a day to work on this project and received one hour of fine arts credit in Advanced Computer Graphics. This contiguous working time allowed the students time to create and draw large segments of the animation. The accomplishments of the summer program laid the essential foundation for the completion of the project.

As the fall semester of the 1993-94 school year commenced, student schedules were rearranged so that most of the corporation could have one class period a day to work together. The interaction of students sharing work assignments required that they be in the computer lab in the same time block.

Designing and rendering computer animation is an extremely time consuming process and is difficult in the framework of a standard school schedule. Students volunteered many Saturdays and evenings mastering animation software, producing storyboards, and drawing frame-by-frame animation on the computer.

As the students echoed frequently, computer capability was a continuous design challenge. Performance and quality began to decrease because within the school budget we were unable to purchase high performance computer animation hardware and specialized equipment. High-grade multi-media animation and rendering requires tremendous amounts of computer memory and storage space. This problem was partially solved when Apple Computer, Inc.® was invited to join the partnership and lend the students the high-end computer resources necessary to complete this project.

IMPACT OF THE PROGRAM

The Project Manager's Perspective:

At the initial meeting, in April 1993, the project students from Edgewater High School were not initially impressive. They listened to the DIS presentation with polite interest. They seemed like a typical cross section of high school students and DIS was not as interesting as skateboarding, track, or music. However, after two weeks, the storyboard presentation for the animation was amazing. The students had researched the concepts of DIS, formed their corporation structure, sketched and colored a storyboard, and constructed a script for the two characters.

Over the next few months, I worked directly with the students through Saturday workshops on the animation software, visits to the school, and telephone calls. Initially the students wanted to give-up at every hurdle; however, slowly, as they became aware that problems would not be simply solved by me, their teacher, or some other outside force, they began to research the issues, work through some of the problems, and decide on alternative plans when a problem could not be solved completely. This group of students gained a new perspective on learning and worked through many of the corporate problems which we deal with on a daily basis.

Teacher's Perspective:

Attempting to produce a high-tech product for a client on a deadline, with neither expertise nor equipment was both challenging and frustrating, but throwing out traditional educational lecture, demonstration and testing methods was a joy!

We are all best motivated by what we see as valuable and applicable to our own lives. Providing students with the opportunity to structure their own goals, through a hands-on project that they have helped design, definitely raises the learning curve - and not only for the student. We know that solid new curricula must be designed to incorporate the best foundations of the past, yet be responsive to what we find crucial for the future--that means re-evaluating not only what is taught, but also how it is learned. Validating assessment of learning (non-traditional grading) and managing special travel and working arrangements for students outside the routine classroom setting required the commitment of everyone involved.

The wide variety of skills these students learned are every bit as valid as the "time in seat" skills of traditional education. The corporation format we used allows students to master skills in a real-world setting, using the expertise of a wide variety of professionals in our community. These students consistently rose to meet challenges that would have frustrated professionals in the graphic arts world and became individually and collectively responsible for their actions.

CONCLUSIONS

When we began this project we never anticipated the intense amount of effort which would be required to bring it to fruition. The time and equipment requirements of this type of multi-media production are foreign to today's public school curriculum. The collaboration needed to utilize experts from throughout the community requires a new attitude about volunteerism and innovative structuring.

As a prototype program, the Instructional Animation project encountered unforeseen problems. Many aspects of the project were more difficult because the technical resources were not readily available. Better access to necessary resources will greatly benefit future projects. Secondly, the visual arts lab was a multi-purpose lab used thoughout the day. For future projects it would be helpful if a dedicated, open-access lab space could be provided so that students could utilize other free periods of the day to continue unfinished tasks. The continuous support of Orange County Public Schools and Edgewater High School administrations dramatically aided this project in overcoming many obstacles.

The benefits derived by the students from this project are basically a result of their active participation in the teaching and learning process. A tremendous amount of the knowledge and skills which the students gained surfaced on a need-to-know basis as the skills became necessary to the students' effort. The students then saw their direct application and remembered the concepts. Additionally, a marked increase in self esteem was noted with each incremental success. Finally, the students gained a respect for the abilities of their peers and the school gained a respect for these students.

Even though the TSD benefits directly from the use of this product, there is a far greater gain. These students are the decision makers and problem solvers of tomorrow. The abilities that they carry from this project will benefit us all.

LAURELS

The students have been recognized for their work by the Organization for Parents and Educators Networking (OPEN) and the Orange County School Board. In addition the project was chosen for presentation at the 1994 World Hyper- and Multi-media Conference, Vancouver, Canada and for publication in the annual School report of Orange County to the State of Florida.

The Naval Air Warfare Center Training Systems Division was also awarded Government Partner of the Year for this project by the 1993/94 Orange County Partners In Education Committee.

REFERENCES

1. W.J. Clinton, A. Gore, Jr., *Technology for America's Economic Growth, A New Direction to Build Economic Strength,* February 22, 1993

2. Institute for Simulation and Training, *Distributed Interactive Simulation Standards Development, Operation Concept 2.3,* IST-TR-93-25, September 1993

ACKNOWLEDGMENTS

We gratefully acknowledge our "Bird dog", Mr. James E. Jardon, II at *JHT Multimedia for his* support during the refinement of this paper.

PROVIDING MILITARY OCCUPATIONAL TRAINING USING COMMUNITY COLLEGES and VIDEO TELETRAINING

Neill H. Foshee
University of Central Florida, Institute for Simulation and Training
Orlando, Florida

Barbara L Martin, Ph.D.
University of Central Florida
Orlando, Florida

Research Sponsored by
Department of Defense Manpower Data Center, Training and Readiness Evaluation and Analysis Division

The need for increased training has prompted the military services, industry and academia to research several different distance education strategies (i.e. courses of instruction packaged for delivery at remote locations), including video teletraining (VTT). Two of the key reasons the military is exploring new methods of distributed training are the size and importance of the reserve components (RC) and continuing reductions in military training budgets. Since RC personnel are only available for an equivalent of 48 training days a year, less expensive, more accessible training methods must be found for reclassifying RC personnel in their occupational specialties.

The purpose of this research effort was to assess the feasibility of using two-year community colleges to offer military courses to RC and active component personnel using a two-way audio and video teletraining system. Five courses were reconfigured for delivery on the U.S. Army Teletraining Network (TNET). Three U.S. Army Reserve Component Configured Courses (RC^3) and two U.S. Navy special topics courses were presented during a four month period in late 1992 and early 1993.

The courses were evaluated on the basis of student performance on standard military proficiency tests and 40 other data gathering instruments. The research demonstrated that VTT is a reliable and effective means for delivering training to military personnel. The VTT approach appears to be acceptable to both students and instructors. Furthermore, the results of the quantitative and self-report data indicate that the VTT instruction was successful in helping students master the learning objectives. The findings also support the premise that community colleges can effectively develop and deliver occupational training to the military.

About the Authors

Neill H. Foshee is a Military Training Specialist at the University of Central Florida's Institute for Simulation and Training. His research interests include military courseware design and evaluation for distance learning, collective training performance assessment tools and cost and training effectiveness analysis. He has been a member of the Florida Army National Guard for the past 13 years and currently holds the rank of Captain. He holds a B.S.B.A. from the University of Central Florida and is currently working on an M.A.. in Instructional Systems at the University of Central Florida.

Dr. Barbara L. Martin is a Visiting Associate Professor in the Department of Educational Services at the University of Central Florida (UCF). Her field of specialization is educational technology and educational psychology. She was formerly the Principle Investigator for the Florida Teletraining Project at the Institute for Simulation and Training at UCF. Prior to going to UCF, Dr. Martin was an Associate Professor and Coordinator of the Educational Technology Program at Kent State University in Ohio.

PROVIDING MILITARY OCCUPATIONAL TRAINING USING COMMUNITY COLLEGES and VIDEO TELETRAINING

Neill H. Foshee
University of Central Florida, Institute for Simulation and Training
Orlando, Florida

INTRODUCTION

Training is the military's primary peacetime mission and the cornerstone of combat readiness. Enhanced training technologies and resource reductions make it logical for the military to examine significant changes in the way it conducts peacetime training in order to realize the greatest value for every training dollar that is spent (U.S. Army DAMO, 1989).

It is estimated that the Reserve Components (RC) constitutes more than half of the Army's combat arms units and more than two-thirds of the Army's combat support and combat service support units (TRADOC, 1990). The military has identified special issues related to individual training in the Reserve Components (RC). These special RC needs generally fall into one of three major areas: (a) job skill enhancement, (b) regular academic or technical programs directly corresponding to a military skill, and (c) specific military skills that civilian providers, such as community colleges, have the resources to design and develop as special offerings (Watt, 1988).

The primary purpose of the Florida Teletraining Project was to assess the feasibility of using two-year community colleges to offer military courses to military personnel using a telecommunications network. The project was charged with determining the extent to which civilian providers could design, develop and deliver specific types of military instruction at a distance.

Advantages of Video Teletraining (VTT)

The need for increased training has prompted the military services, industry and academia to research different distance education strategies. Two of the key reasons for using a distributed training strategy are (a) the importance and size of the RC and (b) dwindling resources (TRADOC, 1990). RC personnel must be trained to the same standards as their active counterparts, yet the RC does not train on a daily basis. Training must be developed to meet both the Army's needs and the time frames available to train RC soldiers. The primary advantage of video teletraining is that it enables the military to provide live, interactive instruction to learners when and where they need it.

Role of Community Colleges

The primary mission of community colleges is to offer viable educational opportunities to all members of the local community. Part of this mission is to provide educational outreach to special segments and groups within the community. Military students are part of the community and community colleges should be able to provide some forms of military training. There is evidence that community colleges with the greatest number of innovations will be the most successful providers of military training (Watt, 1988).

METHODOLOGY

Subjects

The Department of Defense Manpower Data Center, Training and Readiness Evaluation and Analysis Division (DMDC/TREAD) coordinated the selection of the students for this project. Students were selected for the training by their respective military commands, based on normal soldier training requirements. That is, subjects were selected from the normal training population. A total of 275 students were trained during the project. All four services (including the reserve components of the Army and the Air Force), and the U.S. Coast Guard were involved.

The average age of all the students was 33.37 years. A grade of E5 was selected as an approximate estimate of the numbers of students who were managers versus those who were first line supervisors and enlisted personnel. Approximately 63 percent of the students were E5s or below. In addition, approximately 30 percent of the students had a duty position related to the course in which they were enrolled and 4.9% had a civilian occupation related to the course content. All of the students were high school graduates or the equivalent and 15% had a four-year college degree or more. Approximately 19 percent of the students had previously taken courses taught by television.

Video Teletraining Equipment

TNET, the U.S. Army's Teletraining Network, was selected as the communications technology for the FTP. TNET is a two-way, audio-video transmission medium using compressed digital video technology. Most of the video teletraining equipment is housed in a stand-alone, modular video conferencing system.

Participating Community Colleges

Three community colleges participated in the Florida Teletraining Project. Instruction was delivered from the origination site at the Florida Community College at Jacksonville (FCCJ) to three remote classrooms at Valencia Community College in Orlando (VCC), St. Petersburg Junior College (SPJC) and a remote classroom site at FCCJ. Because community colleges strive to meet the needs of their respective constituencies, they have varying instructional and technical resources. For this project, only FCCJ had the technical capabilities to produce the instruction, but each college had the technical personnel, space requirements, and support equipment that were needed to successfully implement the instruction. Therefore, FCCJ was chosen as the lead community college for the project.

Courseware

A total of five courses were reconfigured for delivery on the TNET system. These courses were of two types: U.S. Army Military Occupational Specialty (MOS) courses designed to qualify personnel in their assigned duties and U.S. Navy special topics courses designed to raise student awareness.

Army courses. Three of the courses were U.S. Army RC^3 Military Occupational Specialty (MOS) courses: 71L10, Administrative Specialist; 76Y10, Unit Supply Specialist; and 95B10, Basic Military Police. These courses were delivered once each to Army National Guard and Army Reserve soldiers who were seeking to be reclassified in these MOSs.

71L10 Unit Administrative Specialist. This course is a single-phase course that can be taught in either an Inactive Duty Training (IDT) Phase or an Active Duty for Training (ADT) Phase (U.S. Army Soldier Support Center, 1991). This course was presented in an ADT mode. It was a 73-hour course and was presented during a two-week block from 17 October to 31 October 1992. The Administrative Specialist is responsible for the routine office administration of an activity. He or she works at various organizational levels throughout the Army, from company through division, installation, or higher headquarters.

76Y10 Unit Supply Specialist. This particular course is a dual-phased (IDT and ADT) course (U.S. Army Quartermaster Center and School, 1989). Only the IDT phase was presented during the project. This phase of the course was 96-hours and was presented during a two-week block from 7 November to 22 November 1992. To receive the MOS, the 76Y10 student must also take the ADT phase. Arrangements for completing the ADT phase were made independently of this project. The Unit Supply Specialist performs unit and organization supply procedures. These include the tasks of request, receipt, storage; and issue and accountability of expendable and durable supplies and individual, organizational and. installation equipment.

95B10 Basic Military Police. The 95B10 course is also a dual-phase (IDT and ADT) course (U.S. Army Military Police School, 1991). Only the IDT phase was taught during this project. This phase of the course was 66-hours and was presented during a two-week block from 5 December to 18 December 1992. The ADT phase was presented separately from the project in the summer of 1993. The 95B10 Military Police (MPs) are soldiers who perform the duties of entry level military police, such as: apprehension and search, patrol and traffic operations, investigations, physical security, and self-defense. In addition, the MP prepares and gathers military police information, reports, and forms.

Navy courses. The two U.S. Navy special topics courses were: Handling Hazardous Waste--

Activity Level (HazWaste) and Total Quality Leadership (TQL). The latter courses addressed joint service needs and were made available to members of interested services and components. HazWaste and TQL were offered three and two times respectively. In addition to being offered at the community college remote sites, these courses were also delivered to remote military classrooms located at Ft. Taylor Hardin, Al. and Camp Fogarty, Ri.

Handling Hazardous Waste -- Activity Level is a U.S. Navy course specially adapted for the FTP from the Hazardous Waste Coordinator course. The FTP one-day course was designed to give hazardous waste handlers the information necessary to make environmentally and personally safe decisions regarding the disposal of hazardous and regulated wastes. This course was offered a total of three different times, twice to the community college remote sites on 27 January 1993 and 5 February 1993, and one additional time on 25 February 1993 to the three Florida community college sites, Camp Fogarty and Ft Taylor Hardin. The HazWaste course topics included a review of pertinent laws and regulations and a discussion of the physical and chemical properties of hazardous materials, the correct techniques for delivering and transferring hazardous materials at the hazardous waste collection site, and pollution and spill prevention.

Total Quality Leadership (TQL) is the U.S. Navy's adaptation of Dr. W. Edwards Deming's approach to continuous quality improvement (Mr. Jim Miller, Total Quality Leadership Curriculum Developer, CNET, personal communication, June 2, 1993). A one-day course was presented to provide an introduction to, and an awareness of, the U.S. Navy's TQL philosophy. This course was offered twice, once to the community college sites on 30 January 1993, and once on 22 February 1993 to the three Florida sites and to Camp Fogarty. The topics covered during the course included the background of TQL, how it is defined by the Navy, basic principles, methods and tools used in TQL, and how the Navy has implemented it.

Project Personnel

This distance learning project was large and complex, from instructional, managerial, and technological viewpoints. The number of relationships between community college instructional and credit issues, military training and certification concerns and technical equipment matters, resulted in several layers of technical and instructional personnel, both civilian and military.

The community college instructional personnel required for the FTP were the VTT instructors, who also acted as the course developers and the Instructional Coordinators (ICs) at the remote sites. The VTT instructors/course developers were community college faculty from FCCJ; the ICs were community college faculty at the remote sites. The primary selection criteria for the VTT instructors/course developers and ICs was content expertise.

Military personnel required for the project were the Military Instructional Assistants (MIAs), who assisted the VTT instructors/course developers in the design and delivery of the courseware and the Military Site Coordinators (MSCs) who assisted the ICs at the remote sites. The primary selection criteria for the MIAs was content expertise. MSCs were used only in the MOS courses. Except in the case of 95B, where on-site MOS qualified instructors were necessary for grading of the psychomotor skills, the selection of the MSCs was influenced by the fact that a remote site, military SME was not desired.

VTT Instructor/course developer. All instructional personnel for the FTP were professors, either full-time or adjunct, at the community colleges. At the origination site, four of the five VTT instructors were faculty members selected from departments closely associated with the course they would be teaching. For example, the 71L10 instructor was a professor of Office Systems Technology. The exception was the 76Y10 instructor, who was a professor of English. Since there were no departments or programs that approximated the content of the 76Y10 course, Unit Supply Specialist, this professor was selected because of his previous course development experience and teaching ability.

Despite having some content knowledge and previous course development experience, the VTT instructors/course developers were neither military content experts nor instructional systems design (ISD) professionals. Those course developers without ISD skills had to be taught basic principles. After learning these principles, they then had to apply them to the military content during the course reconfiguration process (Martin, 1993).

Even though four of the five course developers had content expertise, none of them was fully able to reconfigure the courseware or teach the content without the assistance of a military subject matter expert (SME). As a result, a decision was made to use military SMEs to assist in course design and development, to present some of the course content and to answer military related questions.

Military Instructional Assistant (MIA). The MIAs for the Army MOS courses were instructors from the United States Army Reserve Forces (USARF) school located in Jacksonville. These instructors were senior enlisted reservists who were qualified in the MOS of the particular course that they were assisting.

The MIAs for the Navy courses were instructors selected by the respective commands responsible for each of the two courses. One MIA was a Master Chief from the TQL school at the Little Creek Amphibious Base, Va. and the other MIA was a civilian from the environmental compliance department at Cecil Field Naval Air Station, Fl.

Military Site Coordinator (MSC). The MSCs were also instructors from the USARF schools in Jacksonville and Tampa. In the case of the 71L10 and 76Y10 courses, the MSCs were not qualified in the MOS. In 95B10 the MSCs played and important role in demonstrating and evaluating psychomotor tasks at the remote sites. The primary mission of the MSCs, however, was to maintain mandatory military records, resolve problems with soldier pay and military orders and insure that military discipline and courtesy were maintained.

Community College Staff Training. Training of community college personnel was divided into three categories: (a) training for reconfiguration of the military courses into a VTT format, (b) technical training on the operation of the TNET equipment, and (c) training for the presentation of instruction over TNET. There were four groups of community college personnel who received specialized training in preparation for the design and delivery of the courses:

- the course developers who were also the on-camera instructors
- the remote site facilitators
- technical personnel at the origination site and the remote classrooms
- the graphic artists who produced the study guide and computer graphics used during course delivery.

Training for reconfiguration of military courses. The most extensive and intensive training was presented to the VTT instructors/course developers. None of the community college personnel were instructional developers or had taught VTT courses. In addition, they were responsible for converting military courses from a standard or traditional platform format to VTT. Therefore, they received a series of training workshops that varied in length from several hours to several days.

The first block of training was a two-day workshop entitled *Essential Skills for Television Teaching: There is a Difference*. This workshop focused on how to design interactive learning strategies, prepare on-line and off-line questions for students, develop word pictures, and produce an interactive student study guide. The workshop also described the components and processes needed to modify courses for television teaching and demonstrated how to present a positive image on television.

A follow-up series of ISD workshops were presented to the course developers. Workshop topics included: (a) basic principles of learning theory and instructional design, including how to analyze and write learning objectives and how to use the conditions of learning and events of instruction (Gagne, 1985), (b) principles of media selection and utilization, (c) instructional strategies and methods for VTT, (d) an overview of the TNET system and its instructional capabilities, and (e) an overview of military training, including the Army's requirements for Reserve and Guard training, IDT vs. ADT, and the components of a standard Army syllabus.

Technical training on operating the TNET equipment. The next block of instruction, TNET Workshop I, was presented by the Army Extension Training Directorate's (AETD) TNET personnel. While the primary purpose of this training was intended for the technical personnel, the course developers also were taught the basic functions of the equipment and had the opportunity to practice using the newly installed TNET equipment.

A second TNET workshop presented by AETD, TNET Workshop II, focused on the instructional rather than the technical capabilities of the system. During this workshop, the AETD staff also worked with the graphic artists and production team to assist them in designing and producing instructionally sound graphics for VTT.

Training for the presentation of instruction. The final block of formal staff training, Putting It All Together, was a two-day workshop presented to all project personnel including the course developers, technical personnel and the administrative workers at each remote site. The purpose of this workshop was to: (a) explain the technical and instructional roles and responsibilities of each participant, (b) explain the specific instructional procedures that each participant should follow when implementing VTT using TNET, and (c) provide the opportunity for project personnel to form the instructional and technical teams that would be necessary to implement VTT instruction. Topics addressed in this training included: a TNET orientation, classroom protocols, roles and responsibilities, contingency plans, project evaluation requirements and responsibilities, and network policies and procedures. As a conclusion to this workshop, the instructional and technical teams worked together to solve a series of problems that might arise during instruction.

In addition to the formal training, on-the-job training (OJT), individual and team training was implemented. The on-camera instructors also conducted several instructional practice sessions with the remote sites before presenting the courses to the students. The informal training included:

- A 13-page job aid, The Quick Reference Guide to Course Conversion.
- As the course developers/VTT instructors and the MIAs developed each lesson, they were given feedback and assistance in redesigning the lessons.
- The VTT instructors and MIAs were given in instruction on television and presentation techniques by personnel at FCCJ. As part of this instruction, the VTT instructors and MIAs practiced a variety of instructional strategies and they practiced teaching over the network.
- Approximately ten over-the-network training sessions per course (for the MOS courses) were conducted. During these sessions, all the remote site coordinators and the technicians were present. These practice sessions provided an opportunity for all personnel to refine their project roles.

Technical personnel. In addition to attending the TNET Workshops I and II, the technical personnel also received individual training from the AETD staff on the TNET system when the equipment was installed at each remote site. The technical personnel also attended the Putting It All Together workshop and participated in the OJT.

Graphic artists. The graphic artists also received specialized individual training from AETD on how to design graphics for TNET. After the training, the graphic artists conducted an extensive formative evaluation of the graphics and word pictures. This pilot testing was a form of OJT because there was considerable trial and error to determine what colors, sizes, fonts, and clip art were appropriate for TNET and which graphics were instructionally sound.

DELIVERY of INSTRUCTION

The courses were presented over a five month period from October, 1992 to February, 1993. The Army MOS courses were presented between October and December, followed by the two Navy special topics courses in January and February. FCCJ1 was the origination site for all five courses. The three Army MOS courses were each presented once at the tree community college sites in Florida. The Navy Hazardous Waste course was presented at the three community college sites and at Camp Fogarty, Rhode Island and Ft. Taylor Hardin, in Montgomery, Alabama. The TQL course was presented at the three community college sites and at Camp Fogarty.

Both FCCJ1 and FCCJ2 were located in the Downtown Campus of the Florida Community College at Jacksonville. The remote site, FCCJ2, was located in an upstairs classroom in the same building as the origination site. Students at this remote site did not know that the instruction was being broadcast from the same building. This classroom was adapted for use as a pilot test site and official visitation center for the project.

The remote site at SPJC was located at the Allstate Center that houses the college's Criminal Justice Institute. The VCC remote site was located at the McCoy Center for Business and Industry Services that is adjacent to the Orlando Naval Training Center Annex. The remote site at Ft Taylor Hardin FTH) was located in an Alabama Army National Guard Armory in Montgomery. The remote site in Rhode Island was located at Camp Forgarty National Guard Training Site. There were no trained technicians at the out of state sites. Members of the project staff functioned as the ICs for these state sites.

EVALUATION METHODOLOGY

Evaluation Instruments. There were 40 different data gathering instruments developed by project personnel in conjunction with a training evaluation and research contractor in, Lexington, Ky. In addition, six Army forms were used to collect the test data. While there were 32 basic instruments designed by the project staff, there were different versions of the pretest and posttest for each of the five courses.

Several varieties of data were collected for the evaluation. Typical items include the following:

Dichotomous data, usually "yes/no" responses or "like/did not like."

Ratings on a 3- or 5-point Likert scale, where the highest number is always the most positive response.

Ranking a series of responses (coded so the highest number was always the most positive).

Open-ended questions.

Procedures. A complete set of evaluation forms for each participant (students and remote site personnel) in each of the MOS courses were compiled into a data collection notebook and distributed to each remote site on the first day of each course. Directions for collecting the data were included in the notebooks.

All student and instructional personnel interviews were conducted by the project evaluators. Interviews were conducted and questionnaire data collected according to a predetermined schedule. Specific times were set aside at the beginning and end of each course for data collection.

Data Analysis. The data were coded and entered into a database. SPSS for windows was used to analyze the data.

EVALUATION RESULTS

Percentage of Students Passing the Performance Measures

MOS Courses. Students completed the same performance tests (PTs) that were administered to those taking the standard RC^3 courses. There were four performance measures given for 71L10, six for 76Y10, and 35 for 95B10. A summary of the data, giving the percentage of students who passed all the PTs in each course on the first attempt is i included in Figure 1.

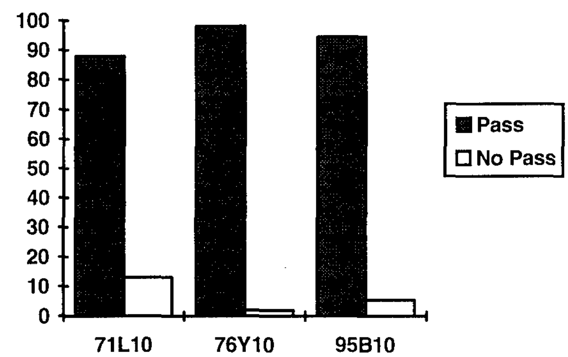


Figure 1. Average Percentage of Students Passing All Tests on the First Attempt in MOS Courses.

Over 85% of the students in the three MOS courses passed all the PTs on the first attempt and 100% of the students passed the course. These results indicate a high rate of success for students in the MOS courses. With the exception of the students who had to retake the typing test (this option was given to all students who did not master typing during 71L10), all students were certified in the course they took.

Special Topics Courses. Students in each of the special topics courses took a 20-item multiple-choice test. These tests were not used for student certification and there were no provisions for retesting. The means and standard deviations are listed in Table 2. For the HazWaste course, the mean score was 78.23, while the mean for the TQL course was 82.90.

Course	Posttest X(SD) (N)
Hazardous Waste	78.23 (10.95) (111)
TQL	82.90 (11.01) (48)

Table 2. Means and standard deviations for Special Topics course posttests.

Students' Acceptance of VTT. A variety of questions were asked to determine the students' feelings toward the effectiveness of the technology and their perceptions of the effect of distance between the instructor and students at the remote sites. In addition to measuring student reactions to the technology, questions related to the students' acceptance of VTT technology were also posed to the MIAs and the MSCs.

Students were asked to rank order their preferred method of receiving instruction, with a rating of one being the least preferred option and two being the most preferred. The mean results from this question are shown in Figure 2. Instruction via VTT at a community college had the highest mean of any other method of instruction, ranging from 3.58 in 76Y10 to 4.01 in HazWaste. VTT instruction at an armory or reserve center had the second highest ranking.

Students were also asked an open-ended question concerning what they liked best about VTT instruction. Of the 79% of the students in all courses who responded to this open-ended question, 23% stated that the technology was the aspect of the instruction that they liked the best. Student responses fell into six broad categories in terms of what they liked best: the instructor, the community college atmosphere, the technology, the course materials, the travel, and other.

When asked an open-ended question about what they liked least about this form of instruction, 79% of the students responded. The percentage who indicated that technology was the aspect of the course that they liked the least ranged from 0% in the 95B10 course to 3% in the 71L10 course. Thus, the data indicate that students were very positive regarding the technology involved with VTT instruction.

Student Ratings of the Instructional Personnel. Teletraining instructors comprise the core of the TNET instructional team, supported at the delivery site by the MIA and the remote sites by the IC and the MSC. Evaluating the effectiveness of this team was achieved primarily from the students' perspectives. Most assessments of the team's effectiveness were not as varied and numerous as those of the VTT instructor.

As in other estimates of the instructional effectiveness, allowance must be made for the novelty of the technology and course design that may have influenced the evaluations of the instructional team members. Students may have responded positively to an instructor or a course simply because they were impressed or excited by the newness of the instructional system. In the evalustion of this project there was no attempt to control for this effect and this presents somewhat of a confound. Further, in conventional instruction, all roles would be consolidated into that of the instructor.

The technology also could have functioned as an obstacle to evaluating the instructor's effectiveness. Students often had a more positive reaction to the IC than to the instructor. This response is perhaps not surprising, since the IC was in the classroom with the students and was able to have more ongoing personal contact. Indeed, the IC was expected to be the instructor's representative in the classroom, providing the student with the personal interaction not available from the instructor through the TNET system. The two-way audio and video offered reasonable student-instructor interaction, but it could not replace the in-class contact found in a conventional classroom setting.

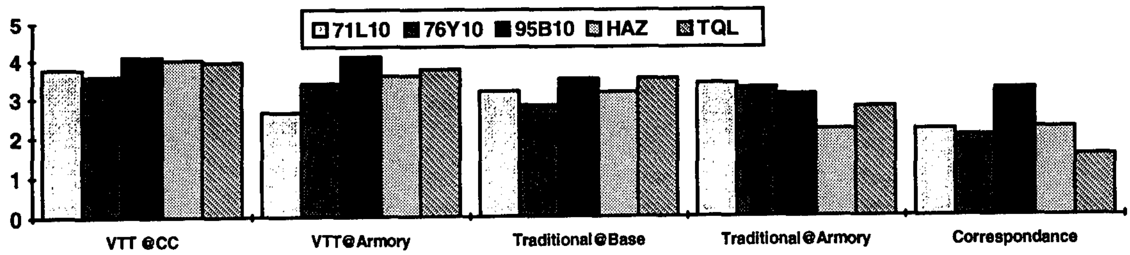


Figure 2. Mean student ratings of various training options.

Just as the technology functioned as the path connecting the instructor to the students, it also served as a barrier between the two, making it more difficult for the students to have a personal relationship with the instructor. One way that the instructor overcame this barrier was through the use of humor and other interactive techniques. Narrative accounts suggested that instructors achieving higher scores on measures of effectiveness were those who appeared to be successful in personalizing the instruction despite the technological obstacles. Another possibility was that those individuals were better instructors in general and the technology had little impact on their techniques. The evaluation did not attempt to control for or analyze these variables.

Student ratings of the Instructor. Students were asked to rate a series of activities on a 5-point scale in regard to how much the activities assisted in understanding the content of the course. The results of the student ratings of the overall performance of the instructors from all five courses are provided in Table 3. Means, standard deviations, sample size and the percent of respondents rating the instructor as very good or excellent (four or five on a five point scale) are presented. The highest rating was given to the TQL instructor, who received a rating of very good or excellent by 89% of her students. Students in 95B10 rated their instructor with an average of almost 4.0 (very good) and 66% of the respondents rated him as very good or excellent

	71L10	76Y10	95B10	HAZ	TQL
X (SD) N	4.09 (.88) 33	4.36 (.90) 39	3.81 (1.10) 26	4.12 (.93) 110	4.48 (.83) 48
% Very good or Excellent	72	82	66	79	89

Table 3. Student Ratings of the VTT Instructor.

Students Ratings of the Military Instructional Assistant (MIA). Students also rated the performance of the MIA. The role of the MIA was to support the civilian VTT instructor at the delivery site: for example, he or she was to answer instructional questions from the military perspective and substitute for the VTT instructor when appropriate. Therefore, results from this item should be viewed with caution since there was a minimum of contact between students and the MIA. At the top of the scale, 85% of the students in both the 71L10 and TQL courses gave their MIA a rating of Very Good or Excellent. However, only 57% of the students in the 76Y10 course rated their MIA as Very Good or Excellent.

Student Ratings of the Instructional Coordinator (IC). Students were also asked to rate how helpful their interaction with the IC was in understanding the content of the course. The ratings of the ICs were higher than the ratings of the instructors on the same question. Ninety-six percent of the students in 71L10 rated interaction with the IC Helpful or Very Helpful in understanding the content of the course, where only 73% of the respondents judged the interaction with the IC to be Helpful or Very Helpful in understanding the content of the course. All but one of the means on this item were over 4.0. Overall, students appeared to consider the IC an important part of the program. As noted earlier in this objective, the narrative data indicated that the student's daily personal contact with the IC led to a more positive evaluation of the IC when compared to the instructor.

CONCLUSIONS

Based on an analysis of the data gathered during the project, the following conclusions about the use of VTT and community colleges to deliver training to the military were drawn:

- Selected military courses can be successfully reconfigured for VTT presentation
- Students indicated that they preferred a VTT approach at a community college to traditional training at a military facility.
- All students passed the stated learning objectives, and over 85% of all students in the MOS courses passed the performance tests on the first attempt.
- Students rated all the instructional personnel, VTT instructors, MIAs ICs, and MSCs, as effective.
- Community college faculty are professional educators. While the faculty at FCCJ lacked instructional design and military training expertise, they were able to design and present high quality instruction given specific training.
- The staff training presented to all instructional and technical personnel enabled them to perform their roles and responsibilities.
- The course developers/VTT instructors required the most extensive training of all the instructional personnel. This included instructional in VTT presentation, instructional design, and military training.
- Course design, development, and delivery required a team approach because the community college faculty did not have sufficient military background. A military SME was needed to assist the community college faculty. This doubling of resources was neither time nor cost effective.

This project demonstrated that, given the time and resources, community colleges can be trained to provide quality instruction to the military using a distance learning network. The question that remains is whether or not civilians should provide this instruction. Some students felt that a military instructor should have presented the instruction, however, all students passed the learning objectives for all five courses. It is hoped that the data from this project will enable policy makers in the DoD and the individual services to develop policies to effectively integrate the use of civilian resources that may be available for military training.

REFERENCES

Cyrs, (1992). *Essential skills for television teaching: there is a difference.* (Available from New Mexico State University, Center for Educational Development).

Gagne, R.M. (1985). *The conditions of learning* (4th ed.). New York: Holt, Rinehart & Winston.

Martin, B.M. (1993). *Reconfiguration of military courses for video teletraining delivery: Florida Teletraining Project Report* Orlando, Florida. Institute for Simulation and Training, University of Central Florida.

TRADOC. (1990). Army training 2001 (Coordinating Draft). (TRADOC Pamphlet No. 350-4). Ft. Monroe, Virginia: Department of the Army, Headquarters United States Army.

U.S. Army DAMO (1989, July). Army Long Range Training Plan 1989-2018. Washington, D.C.: Department of the Army, Office of the Deputy Chief of Staff for Operations and Plans.

U.S. Army Military Police School. (1991, August). *Basic military police course (reserve components) MOS 95B10.* Ft. McClellan, Alabama: U.S. Army Military Police School.

U.S. Army Soldier Support Center. (1991, August). *MOS 71L10: Administrative specialist, skill level 1.* Ft. Benjamin Harrison, Indiana: U.S. Army Soldier Support Center.

Watt, D.M. (1988). *A developing market for continuing higher education: the reserve component* (Report No. Ed 302120). Memphis, Tennessee: State Technical Institute at Memphis.

Computer-Assisted Training in the German Armed Forces

LTC Albert H. Wimmel
Staff Officer GE DOD
Training Technology and Internation pOC NATO
Training Working Group on Training Technology

1. INTRODUCTION

During the last forty years the German Federal Armed Forces have sought to increase training efficiency by using available training tools properly. "Properly" in the context of training refers to the ability to tailor training approaches to meet requirements in a way that is both technically and economically feasible as well as coordinated in methodic and didactic terms.

Today's training environment is one in which resources are becoming scarcer while the amount of training time available is becoming shorter and shorter. Under these circumstances it becomes more imortant than ever to achieve training objectives in a timely and cost-effective manner.

The introductory speech, at the 15th Interservice Industry Training Systems and Education Conference (I/ITSEC) in Orlando, Florida during November, 1993, described the current training of U.S. officers in the following terms:

"The total training period of officers today has changed insignificantly compared to officers thirty years ago. What has changed is the volume of training subjects to be covered. It has more than doubled."

The knowledge explosion situation in the U.S. military is the also present for the German Armed Forces. In order to impart an increased colume of knowledge, more efficient training methods and procedures are required. The German military has opted to introduce advanced training technologies that are capable of putting the emphasis on learning and not teaching. These technologies can make use of idle time and make the learning process more successful and intensive through individualization of instruction to the student's needs.

2. COMPUTER-ASSISTED TRAINING AS A METHOD

Computer-Assisted Training (CAT) technology is a training methodology that is gaining increasing use and acceptance within the German Armed Forces. The goal of CAT is to provide objective oriented and effective training. Given changes in computing over the last 20 years this goal is becoming easier to attain. Early in the computer age, computers were "insider-oriented" machines. They were difficult to use by anyone other than programmers and computer scientists. Attempting to use computers for training, required the learner to first learn how to operate the computer and only then tackle his subject matter. Fortunately, this is no longer as prevalent today. Computers have become more "user-friendly" learning aids which have the capability to adapt to individual learning styles.

At the same time that the user friendliness of computers has increased, the spectrum of their applications for training has been extended greatly. In the German military, whether in a service specific or interservice training context, the question of the applicability of CAT to meeting training requirements is being moe frequently asked. The advantages of CAT were summarized at last year's Interservice Industry Training System and Education Conference by Brigadier General Michitsch, STRICOM Commander, during his introductory remarks. General Michitsch spoke about computers ability to provide instruction which was: individualized; infinitely repeatable with no decrease in quality; time and place independent; and intensive and effective training.

The ability to produce quality instruction is highly dependent upon and limited by the cost and availability of appropriate teams of instructional developers. A CAT development team should consist of a mix of professionals including learning psychologists, education scientists, experts on training objectives and curriculum contents, programmers, media specialists, graphic artists, and importantly evaluation specialists. Teams of these specialists must be capable of anticipating students questions and provide branches in the instruction accordingly. As the applications and capabilities of CAT grow, so to do the demands on the development team. Whether using inhouse personnel or through contracts good personnel to fill development teams are scarce and expensive and are a limiting factor in the growth of CAT in the German military.

The history of CAT in the German Armed Forces goes back twelve years. The first project undertaken was the development of lessons to train operators of the Patriot weapons system. CAT served as a part task trainer. Development of CAT lessons for Patriot was accomplished by a contractor team. The initial product of this work was not satisfactory. We quickly found that leaving the contractors alone to develop the lessons produced results which were not always accurate and not always good instruction. Considerable money and time was required to improve these lessons after they had been handed over to the user. Courseware development in the German military is now produced by the type of multi-faceted team discussed above. Development teams are drawn from the military, universities, and contractors. Generally, psychologists, educators, and evaluators come from the University of the German Armed Forces.

Programmers, media specialists and graphic artists are provided by contract and subject matter experts are usually military personnel.

Development teams have been used to produce courseware for: TORNADO pilot training; electronic reconnaissance; and tactical training for reserve corps officers. CAT courseware has been used in a number of settings. These applications range from distance learning (soldiers training at home) to classroom training of officers or NCOs in schools and academies. We have also worked to ensure that the training that is produced is effective, does it do what it was intended to do. Evaluation of the courseware has been a part of each of our development efforts.

3. AFFECTIVE TRAINING WITH COMPUTERS

The Chief of Staff of the German Armed Forces and the Commanding Officers of the three services have sought to improve the attitudes of soldiers toward their jobs. In response to the Chief of Staff computer-based training (CBT) classrooms were created in three of the military's Noncommissioned Officer Academies. Each academy was equipped with three classrooms. Each classroom had 18 learner stations. Each station was equipped with a 386 SX personal computer which also had a video disc player.

A total of 15 affective lessons were developed by a contractor, university and military development team to teach soldiers to deal with real military life situations. The TenCORE authoring system was used to develop the lessons. Some examples of the types of situations depicted in the lessons are:

- A soldier's immediate supervisor is entangled in a conflict of job demands between his company commander and his squad.
- Integration of a new soldier into a squad in which positions, roles and structures already exist.
- Section leaders having to deal with his soldier's fear during combat operations.

Each of the affective lessons begins with a video depicting the situation which is the subject of the lesson. As noted above one of the lessons teachs NCOs to deal with the fear their soldiers may experience in combat. The video spot for this lesson shows the squad leader and his squad preparing for combat. Both the squad leader and his soldiers have not been in combat previously. The squad leader receives an order from his platoon leader to prepare defensive positions in anticipation of an approaching enemy force. The video makes it clear that combat is imminent. At this point the video stops and the squad leader is presented with alternative courses of action. Based on his decision the video disc branches to show the course of action the squad leader chose and its probable outcome. The squad leaders decisions through out the lesson determine whether a good or a bad solution is reached. The squad leader is able to learn from the consequences of his decisions. In the lesson that teaches about handling fear in subordinates the following training objectives were designed to be met:

- to leave squad leaders with in idea about their responsibilities to their men as a leader in a combat situation.
- to make squad leaders aware that they must constantly attend to what is going on with in their squad to include assessing the feelings and mental state of each individual soldier.
- to learn to identify and handle a number of social situations that could occur within his squad.
- to interpret through their behavior and facial expressions who the most fearful members of his squad are.
- to provide a number of potential "prescriptions" for handling soldier who are experience intense fear reactions.

Using CBT to train NCOs to handle potentially difficult and dangerous interpersonal situations has not been attempted before by the German Armed Forces. We have begun an evaluation of the effectiveness of the courseware. The evaluation is being conducted by two psychologists from the University of the German Armed Forces. Their evaluation program is attempting to answer the following questions:

- How well are the NCOs able to use the computer hardware?
- How easy is it for the NCOs to progress through the lessons?
- Do the NCOs feel that the lessons are depicting realistic situations they have or could encounter?

- Do the NCOs feel that the lessons will help them in performance of their jobs?
- Were the situations depicted in the lessons compelling enough for the NCOs to feel involved in the situation?
- As a result of the lessons, did the NCOs think about how their attitudes had changed?

Approximately 180 NCOs have participated in the evaluation. Half the NCOs received training in the traditional fashion. The other half received the "new" CBT lessons. Both groups of NCOs have been questioned before they began serving as squad leaders, again one year later, and they will be questioned a third time in 1994.
The results of the evaluation are not yet complete.
However, we can say that through the second data collection it appears that the CBT trained NCOs feel better informed and prepared for their jobs. As a result they have better attitudes and have been more successful than the conventionally trained group. More details on the results of the evaluation will be forthcoming upon its completion.

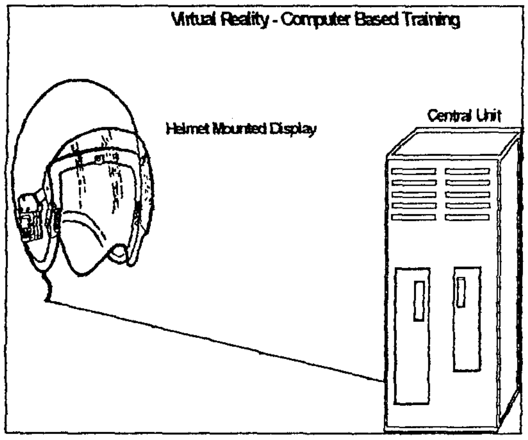
4. COMPUTER ASSISTED TRAINING POLICIES IN THE GERMAN ARMED FORCES

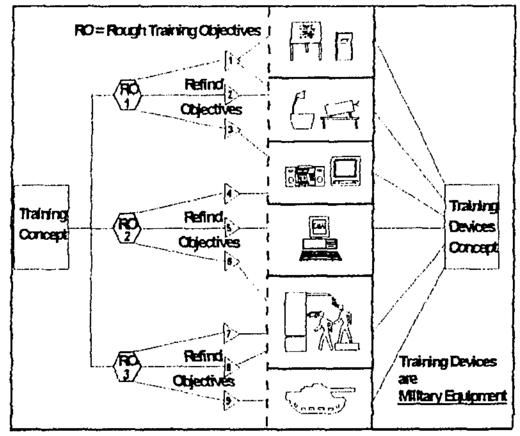
To date, standard practice in the German Armed Forces is for each service to develop its own CAT programs. These programs are service unique and there has been no dialogue or exchange of lessons learned between the services. The services have also not taken advantage of the experiences of non-military institutions. Compounding the problem of each service "going it alone" is the rapid turnover of personnel within the German Armed Forces. This has resulted in needless expense and the wheel being reinvented on numerous occasions. The current policies have proved to be both costly and ineffective.
In order to combat these practices and begin to coordinate CAT activities within the Federal Armed Forces, I set up a working group that first met in December, 1993. The working group is presided over by the Federal Office of Defense Technology and Procurement Technical Affairs Section II 6. This office has responsibility for eliminating needless duplication and waste in the development of CAT training materials. The working group will serve to discuss and disseminate new policies and procedures for the development and utilization of CAT training materials and approaches. Assuming the success of the CAT working group, comparable groups dealing with training simulations and computer-asisted exercises will be formed. Another new working group will be established for "standard training tools". It will deal with issues such as software maintenance, modifications to contracts, writing outline directives, considering hardware follow ups, and dealing with problems of networking.

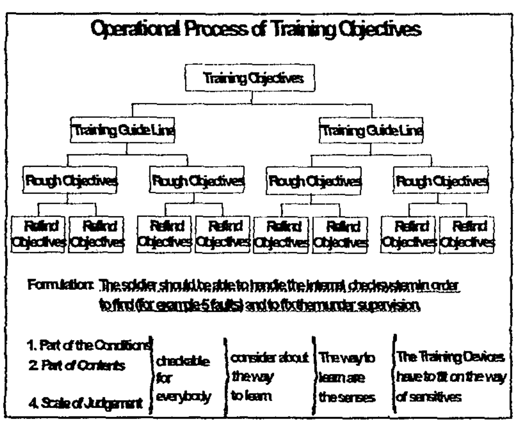
5. CONCLUSION

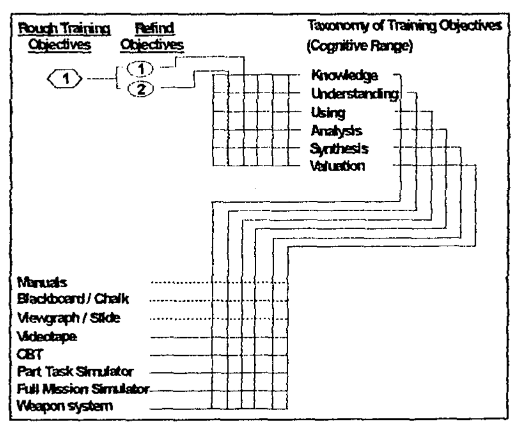
The time of computerized training in the German Armed Forces has just begun. Applications from CBT to full mission simulation are finding a place in German trainig. Environmental protection and scarce resources are forcing our military to find new ways to train. The capabilities of computers to train soldiers continues to grow. Networked simulations allow commanders to experience the complex situations they could face in combat.

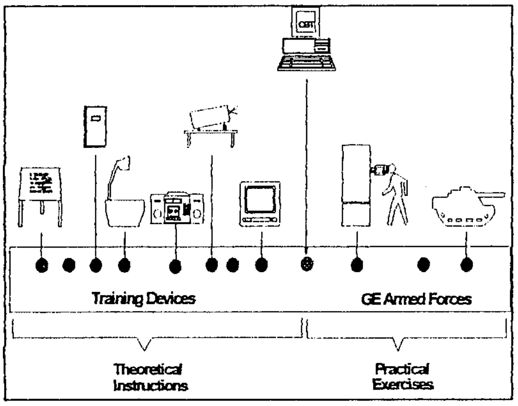
In order to achieve the best possible results in training it is necessary to use the appropriate "tools" for each step along the way. CAT is an obvious choice to impart basic skills and knowledges. The German Armed Forces have been using CAT to train procedural skills for some time. We have just begun to use these methods to train affective skills. We have a ways to go to improve our development process and sharing of lessons learned between services. We have made a start and hope to grow the quantity and quality of our CAT programs over the coming years.

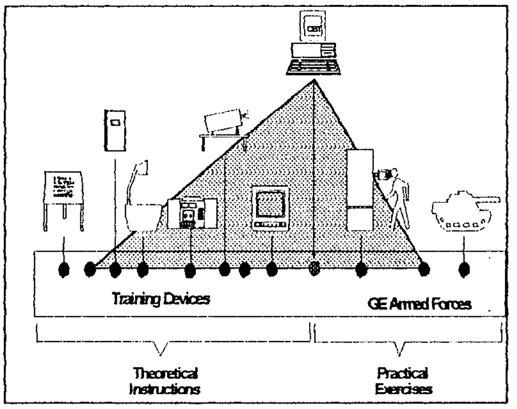

Virtual Reality - Computer Based Training
Helmet Mounted Display
Central Unit

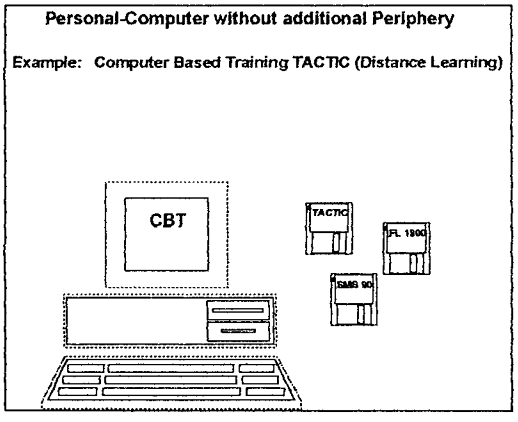

RO = Rough Training Objectives
Training Concept
RO 1
RO 2
RO 3
Refind Objectives
Refind Objectives
Refind Objectives
Training Devices Concept
Training Devices are Military Equipment

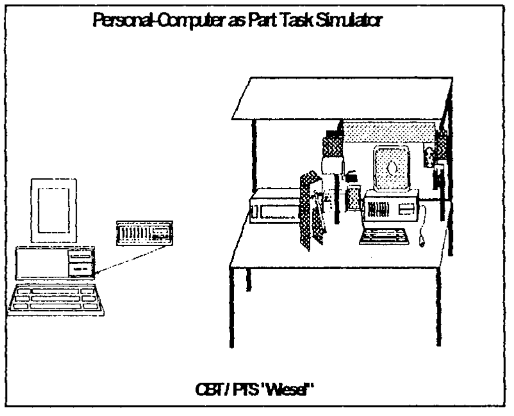

Operational Process of Training Objectives
Training Objectives
Training Guide Line
Training Guide Line
Rough Objectives
Rough Objectives
Rough Objectives
Rough Objectives
Refind Objectives
Formulation: The soldier should be able to handle the internal checksystem in order to find (for example 5 faults) and to fix them under supervision.
1. Part of the Conditions
2. Part of Contents
4. Scale of Judgement
checkable for everybody
consider about the way to learn
The way to learn are the senses
The Training Devices have to fit on the way of sensitives


Rough Training Objectives
Refind Objectives
Taxonomy of Training Objectives (Cognitive Range)
Knowledge
Understanding
Using
Analysis
Synthesis
Valuation
Manuals
Blackboard / Chalk
Viewgraph / Slide
Videotape
CBT
Part Task Simulator
Full Mission Simulator
Weapon system


CBT
Training Devices
GE Armed Forces
Theoretical Instructions
Practical Exercises


CBT
Training Devices
GE Armed Forces
Theoretical Instructions
Practical Exercises


Personal-Computer without additional Periphery
Example: Computer Based Training TACTIC (Distance Learning)
CBT
TACTIC
FL 1900
SMS 90


Personal-Computer as Part Task Simulator
CBT / PTS "Wiesel"

A STRATEGY MODEL FOR COMPUTER BASED TRAINING

William A. Platt & Stephen J. Guynn

Abstract

Improving the state of the art regarding computer based training can be directly linked to the validity and completeness of instructional strategy and the clarity and utility of the terminology and models surrounding the design and development of instructional strategy. Current research emphasis on isolated media variables has not yielded practical results for field practitioners. An alternative holistic approach is to focus on strategy and tactics of computer based training. The **purpose** of this paper is to create a model of strategy and tactics that could lead to a more uniform communications between researchers and developers with categories of strategy that fit the emerging technology. Relevant research issues must be converted into practical guidance of use to designers. Abstract theories must be fortified with working case examples and applications. In a move to operationalize key concepts, four key terms (Interaction, Adaptive, Remediation, and Simulation) were defined in terms of levels of increasing complexity. The proposed model takes into consideration expanded use of artificial intelligence, expert systems, and future use of virtual reality. Learner centered design criteria were identified, with emphasis on interactive formats. The **proposed model** consists of three levels. **General strategy level** consists of a pool of options dealing with the overall training approach. These training approaches can be used in combination to provide a large number of possible general strategies. A sample pool consists of (1) Active Interactive Simulation, (2) Interactive Approximated Simulation, (3) Random Access Discovery Learning, (4) Controlled Path Rehearsal, (5) Scenario Driven Free-play with Active Coaching, (6) Scenario Driven Free-play with Computer Generated Feedback. (7) Opposing Force Game with Active Coaching, (8) Opposing Force Game with Computer Generated Feedback. **Sub strategy (meso tactics) level** deals with the order and use of motivational, evaluative, practice, testing and informational elements. **Working level strategy (basic tactics)** is realized through implementation of a variety of tactics which includes path-option tactics, presentation tactics, learner input or response tactics and feedback tactics. The tactics determine how the audio visual elements will be used as the learner interacts with the program. This is the level that that either makes the overall sequence of events an effective learning experience or a boring, painful and ineffective exercise. A wider selection and mixing of strategy types and tactics along with tighter specification by level of interaction, degree of adaptation, level of remediation, and complexity of simulation; could improve the probability of successful programs. The intended outcome of this model is to provide that opportunity to designers. This will permit Instructional design for CBT to be a flexible exercise, where learning outcomes are more important than rigid formulas for format. The empirical efficacy of various strategies can be established in practice. Training solutions must be evaluated on training effect rather than a tenuous (and often weak) linkage to general theory. Strategy, as used here, should not be confused with theory. Theory must hold over all cases and is therefore general. Strategy bridges the gap from the general to the specific and must only be effective in its intended application. Theory provides guidance and explanation. Strategy leads to accomplishment and the realization of goals and objectives.

Biographical Sketch

William A. Platt received his Ph.D. from Indiana University. He is currently a consultant. Past professional affiliations include Harris Corporation, Loral Corporation and the Naval Training System Center, Orlando Florida. Stephen J. Guynn also received his Ph.D. from Indiana University. He is currently an Evaluation Specialist with the Indiana Prevention Resource Center at Indiana University, Bloomington. Dr. Guynn's past professional affiliations include the Department of the Navy and the Research and Evaluation Department of Indianapolis Public Schools.

A STRATEGY MODEL FOR COMPUTER BASED TRAINING

William A. Platt & Stephen G. Guynn

INTRODUCTION

Instructional innovations have a way of bursting on the scene with high hopes and then fading rapidly into an oblivion of unfulfilled promises, overrun budgets and ambiguous research results. To some extent the history of computer based training or computer based instruction, has followed this course. However, this syndrome is not inevitable. The computer is far too valuable a tool. Computer based training has benefited from a host of second chances. Success stories are mixed in with the disappointments. But the ratio still falls short of the potential. Why has the success rate not been greater? This paper is based upon the proposition that a large part of the answer can be found in three observations on the state of the art. **First,** the language surrounding the art and science of computer based training is full of surplus meaning and is used in differing ways by different practitioners. We have come to depend upon the educational research community for our concepts and terminology dealing with instruction. But instructional research is more often aimed at establishing general principles and conditions. In this endeavor researchers often isolate variables in order to obtain results that will hold up at publication time. Far less research is dedicated to the "praxiology" of instruction where the results are put to use in working settings. That is the domain of strategy and to a great extent it is a neglected domain. **Second,** industry has developed wondrous capabilities to program the computer. In many cases it is the programmer's dream that dominates the basic design and the instructional strategy. Unfortunately most students who use computers are not programmers. This touches on the locus of control issue which has yet to be fully resolved in the realm of CBT. Practical rules for design supported by learning theory or field data, have not kept pace with the programmers capability to invent solutions. **Third,** the available strategy models lack a unifying framework. In some cases models are often closely guarded authoring house secrets. Success in a few areas has limited the selection of strategy to a conservative and narrow path. New workers are "guided" in the company way. This limitation is often compounded by adherence to "templates" which further restricts the creativity and responsiveness applied to program design. Quality control consists of following the pattern rather than measuring the results. These points have recently been supported in the literature on computer based training.

CBT EXPECTATIONS REVISITED

Much of the early hyperbola associated with the tremendous promise of the medium is being set aside in favor of a critical view of the progress made in the last ten to fifteen years. Chen, Brandt, Barbee & Lorenc (1993) point out that CBT is still in its infancy. They recognize that the promise once envisioned for CBT has largely not materialized. Their approach was to focus on improvement by using authoring systems that reduce development time and include the use of a strategy segment library. The use of academic research standards to measure our progress, has turned up some disappointing findings. Terrell (1990) has reviewed the literature on strategies of computer-based instructional design. He was particularly interested in determining the extent that published design guidelines were supported by empirical research. He concluded that very few were supported and that contradictions between guidelines were commonplace as were disagreements between so-called experts. Reeves (1993) casts doubt on much of the learner control research in computer-based instruction. Reeves refers to the research as "pseudo science" which fails to follow the standards of the "quantitative paradigm".

McCombs (1985) investigated Navy and Air Force user acceptance problems with CBT as a function of inadequately prepared instructors and failure to recognize the value of group process roles. A number of other authors have critically examined the theoretical underpinning for aspects of computer training (Milheim & Martin, 1991; Cronin & Cronin, 1992; Rieber & Kini, 1991; Haertel, Walberg, & Weinstein, 1983). Some of our established practices are not supported by empirical research. This of course could partly be because we have been asking the wrong research questions. Rather than teasing out and isolating variables, we should be taking a holistic look at strategy and asking what works and what does not. Rigid academics will undoubtedly continue to follow accepted research designs. Field practitioners on the other hand, face the problem of creating effective instruction and solving training problems under deadlines and cost constraints. Here strategy is useful when it works and discarded when it does not. Both worlds value empirical results but academic research is aimed at establishing grounded theory, while the field practitioner aims at achieving training results. Theory is tested by experimental research, which is linked to the "real" world through operational definitions. To apply theory, some operational variables must be manipulated. In a sense each time a new instructional application is created, a new operational definition is created. In the transition from general rule to specific case, the variation, shading and potential for intervening unknown variables is enormous. Each instance of strategy that claims to follow a principle or a theory must therefore be tested. The role of theory is to generate ideas not to guarantee success. Success is what pilot testing, trial runs, and revising as required are all about. Guidance for practitioners is best expressed as proven strategy rather than general theory.

Emphasis on Strategy. It is time to place emphasis on increasing the inventory of practical strategies that work. Clark (1985) suggests that the training effectiveness of computer based instruction is primarily due to effective instructional strategies and not to the delivery medium. The practitioner uses intuition, hunches, trial and error, and general heuristics, to develop strategy. The point is that these methods often work. When a review of the literature finds that much of current practice is not "supported" by academic theory that does not mean things are not working in the field. In some cases field practice may be ahead of theory. The important thing is to clearly identify what must be learned, and then keep adjusting strategy until the goal is obtained, thus getting back to the original roots of instructional system development. As Kaufman (1992, p 56) points out, we should develop a mindset "that will help us think about how to change what should be changed, keep what works, and modify and develop as required."

CBT STRATEGY ISSUES

What makes up the universe of computer based training / instruction? The "stuff " of CBT consists of instructional issues having to do with learning, delivery issues having to do with the medium, and content issues having to do with analysis and design. Theoretical guidance is evolving (Case & Bereiter, 1984). But the instructional designer must take a stand and use or ignore the guidance currently available. The result is that designer's strategy. This is often invention by default, but never the less it is the designer's attempt to turn ideas into the concrete substance of an instructional program. Designers must consider many things including: Motivation, Reinforcement, Locus of Control, Screen Design, Path Design, Simulation, Motion and Still Frame Video, Animation, Graphics, External Devices, Game Theory, Artificial Intelligence, Pacing, Coaching, Feedback, Team Training, Testing and Evaluation, Remediation and Learning Styles. The point here is that strategy is not a simple thing. Strategy must consider all of this and more and be unambiguously stated. A few commonly used terms have such a range of possibility that they must be pinned down when used in strategy specifications.

TERMS: DEFINITION BY LEVEL

Instructional designers and contracting officers should take note of four terms commonly used for developing CBT strategy. These words are "interactive, adaptive, remediation and simulation." The range of meaning of these terms as they are being applied to computer based instruction is too great to permit them to be used with out accompanying operational definition. The following four tables are intended to show some of the range of meaning possible.

Table 1. Levels of Interaction

Level 1	Learner is exposed to material and advances by manual or automatic "next" switch which is not contingent upon comprehension of material.
Level 2	Learner is exposed to material that requires a decision or discrimination which must be completed. Advancing to the next part is contingent upon a response relative to the content.
Level 3	Learner is exposed to material that requires a decision or discrimination and discrete control input, constructed external response, tool use, or continuos control input.
Level 4	Learner is exposed to material that presents a problem requiring learner to formulate questions, hypothesis, game moves, or dialogue with artificial intelligence, or active player or coach.

Table 2. Adaptive Levels

Level 1	Learner can elect to adjust pacing or other minor control parameters or program makes pacing adjustments based upon response time.
Level 2	Learner can elect to adjust pacing or other minor control parameters or program makes pacing adjustments based upon response time and program includes additional Artificial Intelligence / Expert Systems diagnostics which adjust content, level of detail, and explanation step size.
Level 3	All above plus learner can elect various tracks which accommodate leaning style and program includes a range of voluntary or automatic Artificial Intelligence / Expert Systems diagnostics and helps. For example a discovery learning track in addition to traditional program.

Table 3. Levels of Remediation

Level 1	**Learner fails test or check question and is routed back to earlier point in the same program**
Level 2	**Learner fails test or check question and is provided alternative material that is different from original program either in content, detail, step size, or pacing. Learner re-enters program at point of original failure.**
Level 3	**Learner fails test or check question and is provided diagnostic and then additional material based upon the results and then additional practice before being returned to original point. Artificial Intelligence / Expert Systems used to monitor student may intervene before student make errors to provide additional explanation.**

Table 4. Levels of Simulation (Fidelity, Timing, and Options)

Level 1	**Entity is modeled in part with limited fidelity and not in real time.**
Level 2	**Entity is modeled in multiple functional aspects at moderate fidelity close to real time and limited to visual and audio cues.**
Level 3	**Entity is modeled in full functional aspects in high fidelity in real time with full free play possible , may use virtual reality, and full range of environmental cues.**

A STRATEGY MODEL

The model proposed below takes advantage of the technological options opened up by advances in computer science. The model has three levels, each level focusing on a different aspect of the computer based delivery of instruction.

General Strategy Level. General strategy is the overall specification for action aimed at accomplishing specific goals and objectives. Top level strategy is a statement of the general approach which specifies main feature of the strategy and the active factors in the program. This includes the human participants, the machine capabilities, and the conceptual models that guide the program structure. The number of specific strategies is great, but it is possible to group strategies by salient feature. This creates a pool of base types.

Sub-Strategy (meso tactics) Level. This relates to the use of motivational, informational, evaluative, practice, feedback and testing elements, each containing one or more instructional maneuvers that support the general approach. While the general strategy level indicates the Who, What, When, and Where, of strategy. The sub-strategy covers the all important Why issues. It is where the content meat, the motivational purpose, and the presentation and response interactions are specified. When the elements are added up they deliver a fully functional instructional program.

Working Level Strategy (basic tactics) Level. This level deals with the "How" issue. Each element which serves a purpose related to final goal achievement, must be executed through some tangible means or mechanism. Instructional moves are operatonalized and implemented through tactical employment of various mechanisms for learner interaction with the program. Basic tactics are what the learner experiences when using the program.

Table 5. A Three Level Model Of Strategy and Tactics

The General Strategy Level	**Salient Feature, Participants, Resources**
The Sub-Strategy (meso tactics) Level	**Instructional Moves and Program Parameters, Elements Included,**
The Working (basic tactics) Level	**Mechanics of Elements, Screen Design, Element Path and Linkage,**

EIGHT GENERAL STRATEGY TYPES

Active Interactive Simulation. The key feature of this strategy involves a simulation of an entity which supplies data to the learner who is áble to perform various task operations in any order and receive feed back that reflects operation of the system being modeled. For example a flight simulator or ship simulator can be operated by the learner who can provide control inputs see instrument displays as well as out the window display.

Interactive Approximated Simulation. The key feature of this strategy is that a partial simulation of an entity is use to provide interaction in a sub-set of the system. A panel trainer for example provides correct indications for selected trouble-shooting routines but does not model the entire operation of the system. The learner will only get meaningful system interaction with in the target sub-system.

Database Access Discovery Learning. This strategy involves a relational data base containing some domain of knowledge. The learner is able to query the data base and interact with short learning segments on a random basis. The strategy depends on the learner gradually realizing patterns and relationships with in the data and integrating the data into a learner designed frame-work. Learners may use this strategy in connection to a challenge task that depends upon the data in the data base.

Controlled Path Rehearsal. This strategy is often used in procedure based equipment operation or complex serial tasks like weapons system operation/ deployment or maintenance trouble shooting. The learner steps through the task following prompts that are gradually removed until a practice session is completed free of prompts.

Scenario Driven Free-play with Active Instructor Coaching. When the task involves mission level activities and group behavior that is performed in a team setting, a simulation of the situation may be coupled with an event driven scenario that presents opportunity to practice task interactions under various contingencies which may involve emergencies, unexpected attacks, system failures, or just routine operations under various degrees of stress. Players are free to make moves as they chose in relation to events. Coaches provide feedback, and advice at their discretion. A variation of this strategy can include on line helps and play back elements that can be inserted at the discretion of the coach.

Scenario Driven Free-play with Computer Generated Feedback. This is similar to the above strategy except that the computer is programmed to provide error messages, and other types of feedback in response to certain moves or omissions on the part of the student.

Opposing Force Game with Active Instructor Coaching. This strategy uses game rules tied to the performance of leaning tasks. Learner moves are countered by the coach who may counter learner movers to illustrate some aspect of the task being learned. This strategy is often used to teach students the dynamics of market forces in a competitive environment, or principles of war and weapons employment in military applications. Students may play against each other, against the coach, or against the computer.

Opposing Force Game with Computer Generated Feedback. This is similar to the strategy above except that the computer is programmed to respond to certain situations to counter student moves or provide feedback and advice.

SUB-STRATEGY ELEMENTS

Motivational Elements. A motivational element is a section of a strategy that is intended to motivate the student to learn the material in the lesson. As such the motivational element must be motivating given the learner and the learning task. An example of a motivational element would be a sequence that describes the importance of the task and shows the consequences of failure. Another motivational element would be a sequence that uses a scoring system based upon performance. The element explains that points can be earned toward obtaining a password that would access a higher level skill version of the program. Points would then be awarded as the student performs in the rest of the program.

Evaluative Elements. Evaluative elements are placed in a program to provide feedback on student performance relative to a standard. This form of feedback may be delivered by computer or a live instructor/ coach.

Practice Elements. Practice elements are sequences where the student gets to apply what has just been leaned in an earlier part of the program. The elements may strengthen and widen the learned response being practiced by offering the opportunity to apply the skill or knowledge to a wider set of conditions than the original sequence. Practice elements may be used to test "transfer" to increasingly realistic or difficult situations. Practice elements are usually linked to testing elements or may contain monitoring elements.

Testing Elements. Testing elements check performance. They may be used with information elements or as tests for the record. Testing elements are most often the points of departure for tactical control of the student in the program. Several path options are discussed under tactics. Test elements can be "on-line" or "off-line" as required by the content and nature of the test.

Informational Elements. The content of instruction is delivered in these elements through text, graphics including animation and video and audio voice and tone. In the newer programs that use a "hyper" media format informational elements can also become points of departure to move to other elements using buttons and windows. Informational elements and test elements may be combined into a single frame. It is highly desirable that all advances from informational element to informational element be contingent upon learner comprehension evidenced by a test (see testing element).

On-call and Background Elements. The Monitoring and Pause options are elements that can be called upon at any point in a program. Background elements are used in simulations and other strategies that require background calculations.

TACTICS WITHIN ELEMENTS

Path Option Tactics:

(a) Contingent Moves. The student is required to perform some input action in order to move to another segment of the program. The input may be the answer to a question or recognition of an object etc. but should be meaningful in terms of the content and learning required in the program.

(b) Hyper moves. The student can elect to gain more information about a term or object by selecting it using an input device. The information will be conveyed in a window. The student returns to the program when ready.

(c) Traditional path types. Traditional path types include the **true branch** leads the student to a different concluding point in a sequence. The **side track** leads the student to a common concluding point in a sequence (also called alternate path route). The **return jump** which jumps the student to an earlier point in the program. The **remedial loop** provides additional steps with new material and then returns the student to the point where the loop was joined. The **escape jump** takes the student to a menu or out of an entire sequence. Path types can be used in combination to form larger sets in elements to achieve the element objective.

Presentation tactics: Presentation tactics are the ways that sound, text, graphics animation and video are used to convey information or to elicit a response from the learner. This is one area where creativity and imagination can make the difference between a dull lifeless and boring program and a motivating and effective learning experience.

Learner Input Tactics: Input tactics are the ways that learners can respond to the program. Responses can be discrete selections of point and click answers, constructed keyboard input which is a series of discrete key strokes, or continuos movement as when control inputs are made by using a joystick. Learners may also use test leads to hook up a circuit. Mechanisms like "drag and drop", made available by object oriented programs, have provided some clever possibilities to designers.

Feedback Tactics: Feedback tactics are the ways that sound, (tones and voice), text, graphics, animation, and video are used to confirm, guide, advise, or prompt the student following a student input. Instructor intervention is also used to motivate, correct, adjust, and guide the students performance.

SAMPLE STRATEGY SPECIFICATION

Sample General Strategy: Scenario Driven Free-play with Active Coaching, Training situation: Student Pilot leaning to fly instrument approach following approach plate and air control instructions. Objective: To make the correct control inputs and turn aircraft at the correct points to fly the approach based upon verbal instructions, approach plate and instrument readings. Equipment and software: Computer with interactive level three, remediation level three, adaptive level two programming, level two simulation and mouse and joy stick input devices. The program is free play in that the student can turn the aircraft in any direction not just the correct direction. The program simulates an instrument display and is classified level two because the database and math model are limited instrument readings relative to plane heading and aircraft attitude. The scenario includes the events of the approach in sequence and inputs from an instructor coach who plays the controller.

Sample-Sub-Strategy; Start. Element-1 Motivational, Element 2 orienting instruction, Element 3 cockpit display, Element 4 Explain Control Tower Request, Element 5, Explain come to heading, Element 6, Explain arc intercept, Element 7 Explain turn to final, Element 8 Explain altitude decision On call element error detection alert, On call element program pause/ instructor critique, Background element math model for approach scenario, On-Call element evaluation checks and tests, Student activates scenario by completing tests for elements one to eight. Event control triggered by scenario real time triggers and by aircraft position in the approach. Instructor coach provides feedback as needed.

Basic Tactics: Tactics are required for each element of strategy. Only a few elements from the above example will be shown here due to lack of space. The examples are provided to make the point that the "how to do it" level requires detailed description. Tactics for Element 1(motivational), Fade in- Video sequence of aircraft safely landing Voice comment that the pilot had just completed an instrument approach, Video sequence showing consequences of task failure, Voice comment reminding pilot of the importance of instruments in bad weather. Upon completion of sequence program moves to element 2. Tactics for Element 2 (information) animation of air craft moving through the approach in three dimensional diagram with each stage identified by a pop-up window. Student must answer question to move to element three. Questions are implemented with "drag and drop". Tactics for element 3 (information), Graphic of instruments with highlight of the ones involved, Task statement boxed with objective for lesson. Each of the terms and instruments in element 2 can be used to access a help screen with amplifying information . The student input device for element 2 & 3 is a mouse. Instrument dials are moving according to the heading of a small aircraft model that the student manipulates to trace the approach plate. (These sample tactics statements are only a partial listing for this strategy).

CONCLUSIONS

Improved Specifications. The use of a standard model for stating strategy, and the existence of a data bank of proven strategies, would be of great benefit to instructional developers and to CBT contractors. Statements of work for CBT contracts could include a specification of strategy. Cost estimates can be improved when the complexity of the program is defined using a strategy specification. The nature of CBT is changing. Once the bedrock of a program, the concept of a frame, for example, is no longer straight forward. The simple linking of frames in a path breaks down with the introduction of windows and "hypermedia" formats (Marchionini, 1988; Morariu, 1988). The use of "templates" will require care to avoid uncreative over use of easy fast solutions at the expense of meaningful motivating programs.

Domain Specific. As specific strategies are tested through use in the field, some strategies will tend to be narrow covering only a single target objective. Others may work for a wide range of objectives. Some instructional design texts stress the "domain" independence of strategy. Chen et al. (1993) defines domain as "the combined knowledge and skills in a specific topic area. " From our point of view having a strategy work for more than one domain is desirable, but should not be a design goal for strategy. We encourage creative use of strategy in all domains to see what works and what does not. A failure to try out new ideas is one reason why there are so many examples of dull programs in today's instructional inventory.

Pilot Testing. Greater emphasis on pilot testing and revision of programs and their strategies will ensure that programs are effective. This will also provide researchers with grist for their mills trying to find out why programs work even when no theory can be found to explain the results.

Share Information. Field practitioners must start to share information and results and reach a common usage of terms relating to strategy and tactics. The nature of the exchange between instructional designers and educational researchers also needs to change. Researchers rarely get to ask the question : "why does this strategy work?" As strategy is refined and successful strategy noted, researchers will get that chance. An inventory of proven strategy ideas can be placed in a data based for both field workers and researchers.

A Starting Place. This model is a starting place and should be adjusted each time an innovative new CBT program achieves desirable results. The industry has only begun to explore the possible strategy and tactics combinations. The potential rewards as well as the inevitable frustrations will be even greater as new technology emerges especially artificial intelligence and virtual reality. Will educators and trainers learn from the game designers and entertainment community? Will CBT designers become more creative and adaptive to change? Will clients demand training that is not only informative and accurate but exciting and not-boring? We think yes answers to all three questions are both inevitable and desirable.

REFERENCES

Case, R. & Bereiter, C. (1984). From Behaviorism to Cognitive Behaviorism, to Cognitive Development: Steps in the Evolution of Instructional Design. Instructional Science, 13, 141-158.

Chen, T. T. , Brandt, L. M. , Barbee, D. E., & Lorenc, D. R. (1993). The Global Guide to the World of CBT Authoring. Champaign: Global Information Systems Technology, Inc.

Clariana, R. B. (1993). A Review of Multiple-Try Feedback in Traditional and Computer-Based Instruction. Journal of Computer-Based Instruction, 20, (3), 67-74.

Clark, R. E. (1985). Evidence for confounding in computer-based instruction studies: Analyzing the meta-analyses. Educational Communication and Technology Journal, 33, 249-262.

Cronin, M. & Cronin, K. (1992) A Critical Analysis of the Theoretic Foundations of Interactive Video Instruction. Journal of Computer-Based Instruction, 19, (2), 37-41.

Haertel, G., Walberg, H., & Weinstein, T (1983). Psychological Models of Educational Performance: A Theoretical Synthesis of Constructs. Review of Educational Research, 53, (1), 75-91.

Kaufman, R. (1992). Strategic Planning Plus: An Organizational Guide. Newbury Park: Sage.

Marchionini, G. (1988). Hypermedia and Learning: Freedom and Chaos. Educational Technology, November , 8-12.

McCombs, B. L. (1985). Instructor and Group Process Roles in Computer-Based Training. ECTJ, Fall, 159-168.

Milheim, W. D. & Martin, B. L. (1991). Theoretical Bases for the Use of Learner Control: Three Different Perspectives. Journal of Computer-Based Instruction, 18, (3), 99-105.

Morariu, J. (1988). Hypermedia in Instruction and Training: The Power and the Promise. Educational Technology, Nov. 17-20.

Reeves, T. C. (1993). Pseudoscience in Computer-Based Instruction: The Case of Learner Control Research. Journal of Computer-Based Instruction, 20 (2), 39-46.

Rieber, L. & Kini, A. (1991). Theoretical Foundations of Instructional Applications of Computer-Generated Animated Visuals. Journal of Computer-Based Instruction, 18, (3), 83-88.

Santiago, R. S. & Okey, J. R. (1992). The Effects of Advisement and Locus of Control on Achievement in Learner-Controlled Instruction. Journal of Computer-Based Instruction, 19, (2), 47-53

Terrell, D. J. (1990). Strategies of Computer-Based Instructional Design: A Review of Guidelines and Empirical Research. Technical Report 888 United States Army Research Institute for the Behavioral and Social Sciences: Alexandria, Virginia.

TRAINING DISMOUNTED SOLDIERS IN VIRTUAL ENVIRONMENTS: ROUTE LEARNING AND TRANSFER

Bob G. Witmer, John H. Bailey, & Bruce W. Knerr
U.S. Army Research Institute Simulator Systems Research Unit
Orlando, FL

Kimberly Abel
University of Central Florida Institute for Simulation & Training
Orlando, FL

ABSTRACT

The U.S. Army Research Institute is conducting a research program with the goal of using virtual environments (VE) to train dismounted soldiers. To accomplish this goal, the conditions necessary for transfer of training from VE to real world environments must be identified. This paper reports the results of two experiments investigating the use of VE for training spatial knowledge as it relates to learning routes through large buildings. This task is especially relevant to a hostage rescue situation or other missions performed by special operations forces. Both experiments used the same highly detailed computer model of a large office building. In the first experiment, 60 college students first studied directions and photographs of landmarks for a complex route, then rehearsed the route using either the VE model, the actual building, or verbal directions and photographs. Everyone was then tested in the actual building. Building-trained students made fewer wrong turns and travelled less distance than did VE-trained students, who in turn made fewer wrong turns and took less time to traverse the route than did verbally-trained students. In the second experiment, 64 students practiced a different route using either a landmark-oriented or a left/right direction-oriented instructional strategy, and with their field of view either linked solely to body orientation or controlled by both body orientation and head movements. These data indicate that the use of an instructional strategy that increases the amount of exploration of a VE tends to improve route learning. The use of head tracking, however, had no effect on learning. The results indicate that individuals can learn how to navigate through real world places by training in a VE. While the building model was not quite as effective in training subjects as the actual building, it was much better than verbally rehearsing route directions. The results also suggest that instructional strategy is an important determinant of learning in a VE.

ABOUT THE AUTHORS

Bob Witmer and Bruce Knerr are employed at the U.S. Army Research Institute Simulator Systems Research Unit, located at 12350 Research Parkway in Orlando, Florida. Dr. Witmer is a research psychologist currently involved in a program of research to investigate the use of virtual environments for training dismounted soldiers for Distributed Interactive Simulation applications. Dr. Knerr is a research psychologist and leader of the Simulated Training Environments Team. Dr. Bailey is currently a human factors engineer at the IBM Santa Teresa Laboratory in San Jose, California. Ms. Kimberly Abel is a visual systems scientist at the University of Central Florida's Institute for Simulation and Training. Inquiries should be directed to Dr. Witmer (Phone: (407) 380-4367).

TRAINING DISMOUNTED SOLDIERS IN VIRTUAL ENVIRONMENTS: ROUTE LEARNING AND TRANSFER

Bob G. Witmer, John H. Bailey, & Bruce W. Knerr
U.S. Army Research Institute Simulator Systems Research Unit
Orlando, FL

Kimberly Abel
University of Central Florida Institute for Simulation & Training
Orlando, FL

The Army has made a substantial commitment to the use of distributed interactive simulation (DIS) for combat training, concept development, and test and evaluation. The emphasis in the initial version of DIS (SIMNET) and in the next generation Close Combat Tactical Trainer (CCTT) has been on the simulation of combat for soldiers fighting from armored vehicles, not on dismounted soldiers fighting on foot. Currently DIS does not train dismounted soldiers well, nor does it represent their contribution to the outcome of simulated battles.

We believe that the dismounted soldier can be integrated in the DIS simulated battlefield through the use of virtual environments (VE) technology. Our goal is to determine, through a comprehensive research program, how to best use VE to train dismounted soldiers to perform combat related tasks and to include their contribution to combat outcomes in DIS. To accomplish this goal, the conditions necessary for transfer of training from VE to real world environments must be identified. This paper will report the results of two experiments investigating the use of VE for training spatial knowledge as it relates to learning routes through large buildings. This task is especially relevant to a hostage rescue situation or other missions performed by special operations forces.

VE RESEARCH PROGRAM

The overall scheme for our research program is shown in Figure 1, the Virtual Environment Research Pyramid. The figure shows our research program as a sequential progression from the base to the tip of the pyramid. At the base of the pyramid are task requirements for dismounted soldier training as reported by Jacobs, Crooks, Crooks, Colburn, Fraser, Gorman, Madden, Furness, & Tice (in press). The next level represents previous research in the use of VE for training. Only a few published studies discuss empirical findings in using VE for training (Regian, Shebilske & Monk, 1993; Knerr, Goldberg, Lampton, Witmer, Bliss, Moshell, & Blau, 1993; Kozak, Hancock, Arthur, & Chrysler, 1993). The third level of the pyramid represents four experiments that investigate psychophysical and psychomotor capabilities of observers performing simple tasks in VE. We reported the results of two experiments at this level at the 15th I/ITSEC (Knerr at al., 1993). Research at the fourth level of this pyramid includes two experiments that address the use of VE to teach spatial knowledge, particularly the configuration of and routes through large buildings. The procedures and findings of these two experiments are the subject matter of this paper. At the fifth level, we will evaluate the use of VE to represent exterior terrain, both for training land navigation skills and for applying those skills in the conduct of mission rehearsals and combat simulations. Research at the sixth level will involve the use of VE for tasks that require situational awareness, i.e., complex tasks performed in a changing environment, such as searching for a landmark or a moving object. The top level, team situational awareness, explores the same tasks as level six, but requires communications and cooperation among team members.

LEARNING ABOUT PLACES AND SPACES

Regian, Shebilske, & Monk (1993) list two characteristics of VE that indicate its potential value for training: (1) the VE interface preserves the visual-spatial characteristics of the simulated environment; and (2) the VE interface retains the linkage between motor actions of the participant

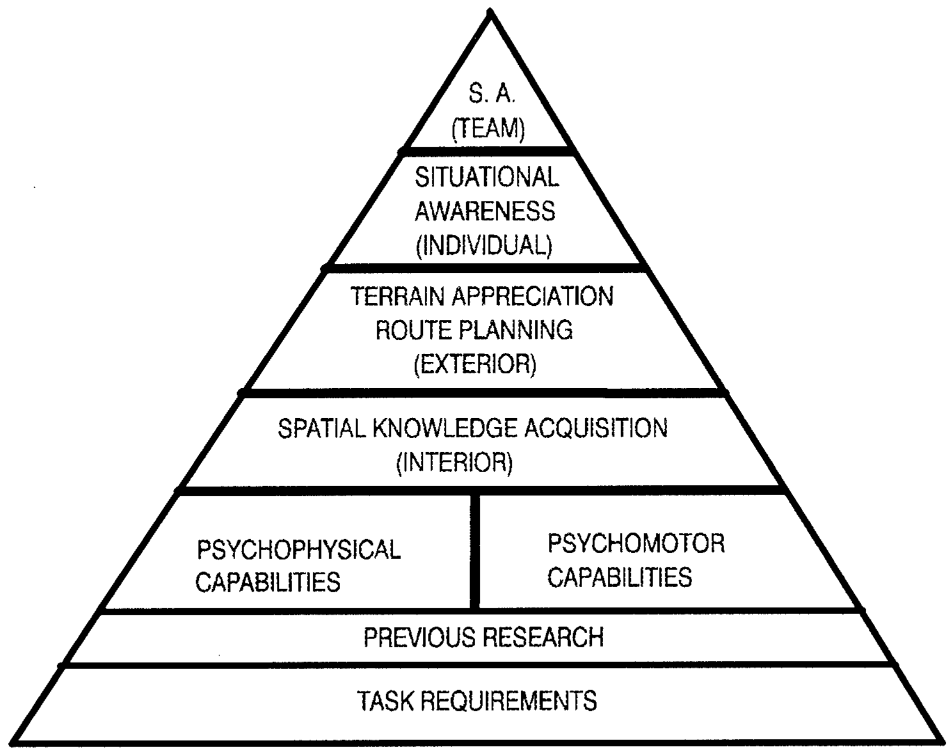


FIGURE 1. THE VIRTUAL ENVIRONMENT RESEARCH PYRAMID

and resulting effects in the simulated environment. These characteristics suggest that VE should be an effective medium for teaching individuals how to find their way around unfamiliar places such as cities or buildings. Because VE preserves the spatial relations and allows you to actively survey the environment through simulated movement and vision, we expect the spatial relationships learned in VE to transfer to the real world environment.

Considerable theorizing and research have been done in order to understand how humans learn to find their way around cities and other complex environments. Nearly a half century ago, Tolman (1948) suggested that animals learned by using a cognitive map. While controversial at the time, Tolman's notion that cognitive maps are instrumental in learning about places is now widely accepted (Lynch,1960; Evans, 1980; Siegel 1981). Siegel and White (1975) suggest that a person's knowledge of spaces generally begins with noticing and remembering landmarks. Landmarks are ".... the strategic foci to and from which one travels" and they help the traveler stay on course (Siegel and White, 1975). Routes linking the landmarks are formed while acting in the context of these landmarks. With sufficient experience in following routes, an overall gestalt of a city, neighborhood, or building may be formed. This gestalt consists of routes and landmarks interrelated in network-like assembly which is or becomes configurational.

A landmark is a unique pattern of perceptual events at a specific geographic location. Lynch (1960) suggests that the number, type, and distinctiveness of landmarks in an environment influences how well individuals can find their way from one place to another in that environment.

Route knowledge consists of the procedural knowledge required to successfully traverse a path between an origin and a destination (Golledge, 1991). It consists of explicit representation of points along the route where turns occur and the actions to be taken at each one. Routes may be learned by associating changes in bearing with landmarks at intersections or choice points (Siegel and White, 1975). The difficulty of learning a route has been shown to vary with the route length, the number of changes in route direction, and the number of route choices at each choice point (Best, 1969). Active exploration of one's environment usually results in the acquisition of routes over a period of time. In some cases, however, routes may be learned more quickly with the aid of maps, written and verbal directions, or both.

The usefulness of using maps, landmarks, and verbal directions for learning about real world spaces has been studied extensively (Canter, 1977; Streeter, Vitello and Wonsiewicz, 1985). The usefulness of these variables in VE, on the other hand, is largely unknown. We performed two experiments to determine the extent to which these variables and others contribute to learning about spaces and places in a VE. The performance of participants trained to follow a specified route in a VE (VE Group) was compared to the performance of participants who were trained in the actual building (Building Group), and to the performance of participants who were trained using only verbal instructions and photographs of landmarks (Symbolic Group). The Building Group and the Symbolic Group served as control groups against which to evaluate the effectiveness of the VE as a training medium. The Building Group was included to determine the best performance that could be expected from naive participants with limited route study and route rehearsals. The Symbolic Group was included to determine how less expensive representations of the building route compared to VE as a training alternative. Half of each group was allowed to study a map in order to evaluate the contribution of map study to learning for each training medium.

MATERIALS

Both experiments used the same highly detailed computer model of a large office building. The University of Central Florida Institute for Simulation and Training (IST) modeled the four-floored building in great detail using Multigen by Software Systems and WorldToolKit by Sense8 Corporation. The completed building model, comprising areas on three floors of the building, consists of over 40,000 flat-shaded polygons, many of which are texture mapped and capable of dynamic behavior. The simulated building was run on a Silicon Graphics Crimson Reality Engine. The model is very rich in detail and includes all of the most prominent landmarks, many of the office furnishings, and many other details including overhead lights, baseboards, and exit signs.

Participants in Experiment 1 used the Fakespace Labs two-color BOOM2 high resolution display from a standing position to view and control their movement through the VE, while seated participants in Experiment 2 used a joystick to move, and viewed the VE through the low resolution Flight Helmet, an HMD designed by Virtual Research.

EXPERIMENT 1. TRAINING TRANSFER

Procedure

In the first experiment 30 male and 30 female participants first studied written directions and photographs of landmarks for a complex route, either with or without a map, then rehearsed the route using either the VE model (VE Group), the actual building (Building Group), or verbal directions and photographs (Symbolic Group). Participants were limited to 15 minutes for reviewing the route study materials. Each participant then rehearsed the entire route three times, with unlimited rehearsal time. Following rehearsal, we tested all participants for their knowledge of the route by asking them to traverse the route in the actual building. Participants were stopped and informed that they had taken a wrong turn each time that they deviated from the prescribed route. The experimenters recorded the number of attempted wrong turns and the total time to traverse the route. Total distance traversed was also recorded using a pedometer.

Results

The primary objective of this research was to assess differences in training transfer as a function of rehearsal mode (Group effect). These differences were evaluated using a Multivariate Analysis of Variance (MANOVA) with Rehearsal Mode, Map, and Gender as the independent measures. Only the main effect for Rehearsal Mode was significant, both overall, $p<.001$, and for each of the dependent measures: route traversal time, $p<.001$; number of wrong turns, $p<.001$; and total distance travelled, $p<.05$. Participants trained in the building made fewer wrong turns, $p<.05$, and travelled less distance, $p<.05$, than did subjects who were trained in the VE. VE participants, in turn, made fewer wrong turns, $p<.01$, and took less time to traverse the route, $p<.01$, than did participants who were trained symbolically. These means are summarized in Figure 2.

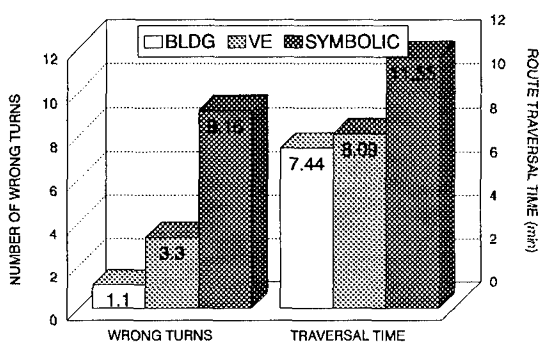


FIGURE 2. WRONG TURNS AND ROUTE TRAVERSAL TIME AS A FUNCTION OF TRAINING MODE

The finding that the VE Group performed significantly better than the Symbolic Group indicates that training transfer from VE to the real world occurred, but small significant differences between the VE Group and the Building Group suggests that the transfer was not perfect. The advantage of VE as a training medium for training spatial skills is clear when you consider that the Symbolic Group made nearly three times as many wrong turns as the VE Group and took almost four minutes longer to traverse the route on the training transfer test. For cases where it is impossible or impractical to train in the actual environment, VE appears to be an excellent alternative.

A look at performance across the three rehearsal trials (see Figures 3 and 4) provides insight about the change in performance for the various training media groups, and may explain why the VE Group did not do as well on the transfer test as the Building Group.

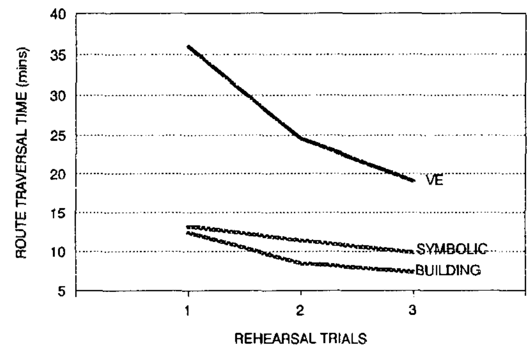


FIGURE 3. ROUTE TRAVERSAL TIME AS A FUNCTION OF NUMBER OF REHEARSAL TRIALS AND TRAINING MEDIUM

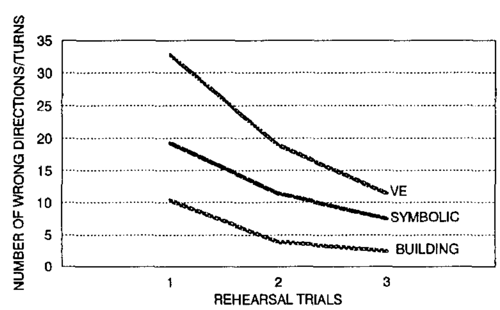


FIGURE 4. ROUTE TRAVERSAL ERRORS AS A FUNCTION OF NUMBER OF REHEARSAL TRIALS AND TRAINING MEDIUM

Route rehearsal times for the VE Group as shown in Figure 3, are significantly slower than the rehearsal times of the Symbolic and Building Groups, $p<.01$, as revealed by post hoc contrasts. Also, the learning curve of the VE Group has a steeper slope than learning curves of the Symbolic and Building Groups, $p<.01$. VE Group rehearsal times decrease across trials at a faster rate.

The differences in rehearsal times and the slope of the learning curves may be attributed to the skill requirements imposed by each of the training environments. Participants in the Symbolic and Building groups were not required to learn any new skills in addition to learning the route. The VE participants, however, in addition to learning the route, were required to learn how to maneuver in the VE using the BOOM2. Learning how to negotiate a winding stairway and how to maneuver away from walls after a collision using the hand controls on the BOOM2 may account for the slower rehearsal times and steeper learning curve observed for the VE Group.

This experiment answered several questions regarding the effectiveness of VE for teaching individuals about places. It clearly demonstrated that navigation skills learned in a well-designed VE transfer to the real world. It also showed that some characteristics of today's VEs can slow the course of learning when compared to training in real world environments.

However, this experiment left many issues unresolved. For example, is it necessary to use a high resolution display device to train routes through a building as was the case in this experiment or might a lower resolution device be as effective? Is it necessary to couple head movements to a changing view for effective training or might the same result occur using a joystick to "look around"? Finally, can the amount of learning in a VE be increased by instructions that are designed to increase exploration of that environment?

EXPERIMENT 2. INSTRUCTIONAL STRATEGY AND CONTROL

Procedure

In the second experiment, 32 male and 32 female participants rehearsed a circuitous route in the VE using an instructional strategy either based on following successive landmarks (exploratory instructions) or following left/right style directions (restrictive instructions). The attention of participants who used the landmark-based strategy was directed toward paintings on the wall or to other landmarks strategically located at the intersection of hallways. Participants' field of view (FOV) was either linked solely to body orientation (controlled by joystick manipulation) or controlled by both joystick manipulation and head movements (i.e., coupled to head movements via a head tracking device). Following rehearsals, all participants completed route knowledge and building configuration knowledge tests. Route knowledge was measured in two ways: (1) by recording time, attempted wrong turns, and distance traveled as participants traversed the route using a joystick and CRT display; and (2) by recording each participant's score on a route photograph ordering task. For the latter measure, participants placed a series of randomly ordered photographs taken along the route in the actual building in the correct order.

Results

Photograph Ordering Test. A 2 x 2 between subjects analysis of variance was performed on the photograph ordering test data. A participant's score was the rank-order correlation between the participant's ordering of the photos and the true photo order. The independent variables were instructional strategy (exploratory and restricted) and head-tracking (tracking and no tracking).

The only significant effect was instructional technique, $\underline{p}<.05$. The exploratory instruction group ($\underline{M}$ = .68) had significantly higher correlation scores than the restricted instructions group ($\underline{M}$ = .57). This indicates that the exploratory instruction resulted in better recognition of real-world photographs and superior ability to place the photographs in order as they occurred along the route.

Route Traversal Test. Because the raw data for the route test did not follow a normal distribution, a natural log transformation was used to normalize the data before performing the statistical analysis. A 2 x 2 between subjects multivariate analysis of covariance (MANCOVA) was performed on the transformed data for the number of wrong turns and traversal time. The independent variables were instructional technique (exploratory and restricted) and head-tracking (tracking and no tracking). The covariates were participants' scores on a test of spatial ability (paper-folding test) (Ekstrom, French, Harmen, & Dermen, 1990) and their reported confidence in using computers.

A Multivariate Analysis of Variance (MANOVA) showed that combining the number of wrong turns with route traversal time yields a significant effect for instructional technique, $\underline{p}<.01$, but not for head-tracking. There was no significant interaction.

Instructional technique significantly affected route traversal time, $p<.05$, but not the number of wrong turns. Participants who had exploratory instructions traversed the route significantly more slowly (M = 4.85 min.) than participants who had restricted instructions (M = 4.29 min.). These results are opposite to what one would expect if the exploratory instructions had resulted in superior learning of the route (e.g. one would expect them to move smartly through the route without lingering or hesitating). It is possible that the exploratory participants were more cautious or deliberate in traversing the route. This possibility is supported by the fact that the number of wrong turns made by the exploratory group was less than the number made by the restrictive group; however, as noted above this difference was not statistically significant.

Route Traversal Test Without Sick Participants. The route traversal test data was reanalyzed using MANCOVA with the data from any participants who had experienced moderate or severe simulator sickness symptoms removed. The combined number of wrong turns and route traversal time was significantly affected by instructional technique, $p<.01$, but not by head-tracking. An investigation of the means revealed that the significance of the combined variables was likely due to a trade-off of route traversal time and wrong turns. Participants who had exploratory instructions traversed the route more slowly (M = 4.69 min.) than restrictive participants (M = 4.23 min), but the exploratory participants made fewer wrong turns (M = 3.21) than the restrictive participants (M = 4.08). The results are shown in Figure 5.

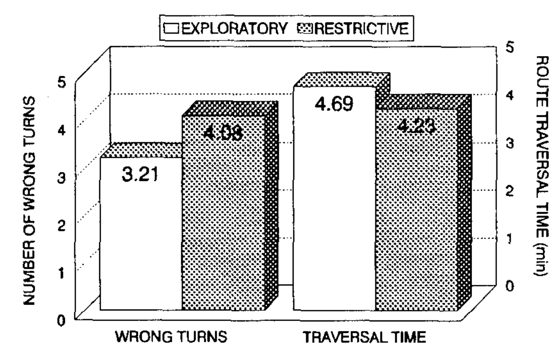


FIGURE 5. WRONG TURNS AND ROUTE TRAVERSAL TIME AS A FUNCTION OF INSTRUCTIONAL STRATEGY

MEASURING SIDE EFFECTS

Simulator Sickness

In both experiments we administered a self-report measure of simulator sickness, the Simulator Sickness Questionnaire (SSQ) (Kennedy, Lane, Berbaum, & Lillienthal, 1993). The SSQ measures three dimensions (Oculomotor Discomfort, Disorientation and Nausea), each consisting of several related factors that represent symptoms associated with sickness in simulators, as well as an overall Total Severity score. Symptoms include eyestrain, difficulty focusing, blurred vision, headache, dizziness, vertigo, nausea, stomach awareness, salivation and burping. Knerr, et al. (1993) have shown that VE can produce significant simulator sickness that may exceed that produced by standard aircraft simulators.

Figure 6 shows the simulator sickness profiles for participants who completed each of the two experiments. Four of 24 VE Group participants in Experiment 1 and 11 of 75 participants who started Experiment 2 could not complete the experiment because of simulator sickness. Participants who dropped out appeared to have much higher Nausea scores than those who did not.

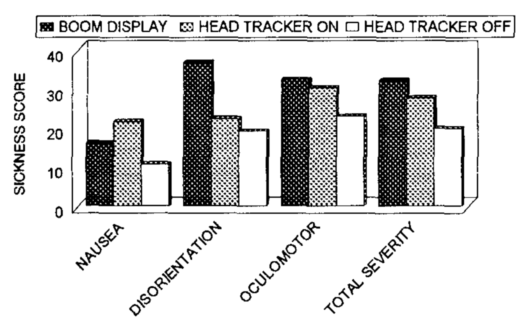


FIGURE 6. SIMULATOR SICKNESS SUBSCALE AND TOTAL SCORES

The Total Severity scores were higher in Experiment 1, probably because of the longer exposure to the VE, but also because the head movements of half of the participants in Experiment 2 (those participants without head tracking) did not change their field of view. In Experiment 2, the group with head tracking (M = 21.76) reported significantly more Nausea, $F(1, 60) = 4.01$, $p<.05$, than the group without head tracking (M = 10.74). Also, of the 11 participants who were unable to complete Experiment 2 due to simulator sickness, eight were in the head tracking condition and three were in the no head tracking condition. Note that the group using the Flight Helmet with head tracking experienced slightly more Nausea than the participants who had the BOOM2 display, despite longer exposures for the latter. Greater lags between the initiation of movement and scene change for the participants using the HMD with head tracking, coupled with more head movement, may be responsible for the differences in Nausea among the groups. Another difference that might account for the higher Nausea scores in the group who had head tracking was that some frames were intentionally dropped out to reduce the perceived lag that would otherwise occur when the participants quickly turned their heads.

Presence

Presence may be defined as the subjective experience of being in one place when you are physically in another (Witmer & Singer, in press). The amount of presence experienced in a particular environment may depend on a number of individual and environmental factors, including the degree, immediacy and naturalness of control experienced by the user, the degree to which the user perceives movement, consistency of information across modalities and with the objective world, attention to external distractions, and ability to modify the physical environment (Sheridan, 1992; Held and Durlach, 1992). Witmer and Singer (in press) have developed a questionnaire, incorporating these factors and others, to measure presence in VEs. We administered this Presence Questionnaire (PQ) to participants in both experiments following their exposure to the virtual building model. The mean presence score reported in Experiment 1 using the BOOM2 device was $\underline{M}$ = 144.55. The presence scores reported in Experiment 2 using the Flight Helmet differed slightly for those participants who had head tracking ($\underline{M}$ = 143.84) and those who did not ($\underline{M}$ = 139.63).

Experiment 1 and Experiment 2 PQ scores were significantly negatively correlated with Simulator Sickness scores, $r = -.60$, $\underline{p}<.01$, and $r= -.35$, $\underline{p}<.005$, respectively. This finding is contrary to the prediction of researchers (e.g., Kennedy, Lane, Lillienthal, Berbaum, & Hettinger, 1992) who equate high levels of presence to increases in simulator sickness. Consistent with our finding, one might expect participants who focus on feelings of discomfort due to simulator sickness to be less immersed in VE than someone who is not feeling sick and can concentrate more on other aspects (e.g., images, sound, task characteristics) of the VE.

In Experiment 2, neither head tracking nor type of instructions had a statistically significant effect on the amount of presence reported on the PQ. The additional simulator sickness experienced by the participants who had head tracking may have moderated the differences in presence that might be expected as a function of head tracking. The mean values of presence reported in the two experiments were nearly equal, indicating that the type of display used was not a strong determinant of presence.

DISCUSSION

In a recent movie, VEs were portrayed as presenting information in a way that resulted in very rapid knowledge acquisition. The VE was so effective that a character in the movie was transformed from a simpleton to a genius in a matter of months. In reality, there is no evidence to suggest that learning occurs more rapidly in a VE than it would in the real world. Knerr et. al. (1993) have presented data that show that performance of psychomotor tasks trained in a VE improves with additional practice in that environment. While Regian, Monk, & Shebilske (1993) have provided some evidence that real world skills can be trained in a VE, Kozak, et. al. (1993) were unable to demonstrate transfer from the VE to the real world. Regian, Shebilske, & Monk (1993) compared the effectiveness of using a 2-D "God's eye view" of a building for training configuration knowledge with a virtual reality representation of that same building. Tests of navigation in the real building tended to favor the 2-D representation, but the differences in the two training conditions were small. Regian, however, did not compare the effectiveness of VEs with a real world environment as a training medium. And previous work has done little to identify the conditions that influence learning in a VE.

Experiment 1 clearly demonstrates positive training transfer from a VE to the real world, and also shows the effectiveness of the

VE as a training medium compared to the real world environment. Experiment 2 shows that instructions that encourage exploration may enhance learning in a VE. In addition it was clear that route learning occurred in Experiment 2 despite the poor resolution (approximately 16 arc minutes per pixel) of the Flight Helmet.

Both experiments support the observation that VE can produce significant simulator sickness. Note that simulator sickness occurred despite differences in type of display device (head-mounted vs boom-mounted) and body posture (sitting vs standing). Total Severity scores and scores on two of the subscales were higher in Experiment 1, possibly due to the longer exposures to VE in that experiment. Nausea seems to be less affected by length of exposure, and participants who experience significant Nausea often report feeling nauseous in the first few minutes that they are in the VE.

The amount of presence reported in Experiment 2 was about the same as reported in Experiment 1 despite differences in control and display devices. The amount of presence reported was slightly less for participants who did not have head tracking, which suggests that presence may be affected by that factor.

IMPLICATIONS FOR DISMOUNTED SOLDIER TRAINING

This research has demonstrated that spatial skills learned in a VE transfer to the real world. Thus, we may create virtual models of enemy terrain or other strategic sites, and dismounted infantry can learn about the terrain or site without ever having set foot on enemy territory. This will allow our soldiers to rehearse a mission without compromising their safety or security. We have seen that VE incorporating low resolution displays can train effectively, and that spatial learning without head tracking may be just as effective as learning with head tracking, and may produce less simulator sickness. It remains to be seen whether systems with head tracking that produce less lag are more training effective than are systems that do not incorporate head tracking.

IMPLICATIONS FOR FUTURE RESEARCH

When the task is to learn routes, configurations, and other spatial skills, it should not be necessary to do a separate transfer study each time that a new VE is developed if the following conditions exist: (1) the VE being considered is a reasonably close approximation of the actual environment; and (2) there are not characteristics (e.g., larger lags) of the simulation that would grossly interfere with learning. In conducting future VE training research, it would be wise to remember that VEs that require participants to acquire new skills in addition to the primary task will slow the course of learning in those environments. Researchers might also try to minimize the amount of head movement if the VE under study produces high rates of simulator sickness.

REFERENCES

Best, G. (1969). Direction-finding in large buildings. In Canter, D.V. (Ed.) Architectural psychology. (pp. 72-75). Cambridge: W. Heffer & Sons ltd.

Canter, D. (1977). The psychology of place. New York: St. Martin's Press, Inc.

Ekstrom, R. B., French, J. W., Harmen, H. H., & Dermen, D. (1990). Manual for kit of factor-referenced cognitive tests (Office of Naval Research Contract N00014-71-C-0117). Princeton, New Jersey: Educational Testing Service.

Evans, G. W. (1980). Environmental cognition. Psychological Bulletin, 88(2), 259-287.

Golledge, R. G. (1991). Cognition of physical and built environments. In Garling, T. R. & Evans, W. (Eds.) Environment, cognition, and action: An integrated approach. (pp. 35-62). New York: Oxford University Press.

Held, R. & Durlach, N. (1992). Telepresence. Presence, 1, 109-112.

Jacobs, R. S., Crooks, W. H., Crooks, J. R., Colburn, E., Fraser, R. E., Gorman, P. F., Furness, T. A., & Tice, S. E. (1994). Training dismounted soldiers in virtual environments: Task and research requirements. (ARI Technical Report). Alexandria, VA: U.S. Army Research Institute. Manuscript submitted for publication.

Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). A simulator sickness questionnaire (SSQ): A new method for quantifying simulator sickness. International Journal of Aviation Psychology, 3(3), 203-220.

Kennedy, R. S., Lane, N. E., Lillienthal, M. G., Berbaum, K. S., & Hettinger, L. J. (1992). Profile analysis of simulator sickness symptoms: Application to virtual environment systems, Presence, 1(3), 295-301.

Knerr, B. W., Goldberg, S. L., Lampton, D. R., Witmer, B. G., Bliss, J. P., Moshell, J. M. and Blau, B. S. (1993). Research in the use of virtual environment technology to train dismounted soldiers. Paper presented at the 13th Annual Interservice/Industry Training Systems and Education Conference. Orlando, FL.

Kozak, J. J., Hancock, P. A., Arthur, E. J., and Chrysler, S. T. (1993). Transfer of training from virtual reality. Ergonomics, 36(7), 777-784.

Lynch, K. (1960). The image of a city. Cambridge, Massachusetts: M.I.T. Press.

Regian, J. W., Monk, J. M., & Shebilske, W. L. (1993, July). Virtual reality: Basic research on the effectiveness of training transfer. Paper presented at the 29th International Applied Military Psychology Symposium. Wolfson College, Cambridge, UK.

Regian, J. W., Shebilske, W. L., & Monk, J. M. (1993, May). A preliminary empirical evaluation of virtual reality as a training tool for visual-spatial tasks. (Report No. AL-TR-1993-0004). Brooks Air Force Base, TX: Armstrong Laboratory Human Resources Directorate Technical Training Research Division.

Sheridan, T. B. (1992). Musings on Telepresence and Virtual Presence. Presence, 1(1), 120-125.

Siegel, A. W. (1981). The externalization of cognitive maps by children and adults: In search of ways to ask better questions. In Liben, L.S., Patterson, A.H., & Newcombe, N. (Eds.) Spatial representation and behavior across the life span: Theory and application. (pp. 167-194). New York: Academic Press, Inc.

Siegel, A. W. & White S. H. (1975). The development of spatial representations of large-scale environments. In Reese, H. (Ed.) Advances in child development and behavior (Vol. 10, pp. 10-55). New York: Academic Press, Inc.

Streeter, L. A., Vitello, D., & Wonsiewicz, S. A. (1985). How to tell people where to go: Comparing navigational aids. International Journal on Man-Machine Studies, 22, 549-562.

Tolman, E. C. (1948). Cognitive maps in rats and men. Psychological Review, 55(4), 189-208.

Witmer, B. G. & Singer, M. J. (1994). Measuring presence in virtual environments. (ARI Technical Report). Alexandria, VA: U.S. Army Research Institute. Manuscript submitted for publication.

Virtual Environments in Training: NASA's Hubble Space Telescope Mission

R. Bowen Loftin and Patrick J. Kenney;
Robin Benedetti, Chris Culbert, Mark Engelberg, Robert Jones, Paige Lucas, Sean McRae, Mason Menninger, John Muratore, Lac Nguyen, Laura Pusch, Tim Saito, Robert T. Savely, and Mark Voss

ABSTRACT

Virtual environment (VE) technology was used to construct a model of the Hubble Space Telescope (HST) and those elements that were replaced or serviced during the December, 1993 repair and maintenance mission conducted by the National Aeronautics and Space Administration (NASA). The VE also included the payload bay of the Space Shuttle and the fixtures used for transporting replacement systems into orbit. Beginning in September, 1993, approximately 100 members of the NASA HST flight team received over 200 hours of training using the VE. In addition to faithfully replicating the physical structure of the HST and the interrelationships of many of its elements, the VE also modeled the constraints associated with all maintenance and repair procedures. For the first time, a VE was integrated with a limited capability Intelligent Computer-Aided Training (ICAT) system. The ICAT component of the training provided identification of all relevant features of the HST, monitored procedures carried out by the trainees in real time, and intervened with assistance in response to procedural errors or requests for assistance. Data collected from trainees, after completion of the HST mission, demonstrated that, for most trainees, the VE training enhanced the effectiveness of their job performance. The results of this project serve to define the future role of VEs in training within NASA and to provide evidence that VEs can successfully support training in the performance of complex procedural tasks.

AUTHORS' BIOGRAPHIES

R. Bowen Loftin holds a B.S. in Physics from Texas A&M University and an M.A. and a Ph.D. from Rice University, also in Physics. He serves as NASA/Johnson Space Center's Principal Investigator for Advanced Training Technologies. In his "other" life, Dr. Loftin is Professor of Computer Science and Director of the Virtual Environment Technology Laboratory at the University of Houston and Professor of Physics at the University of Houston-Downtown. Since 1983, Dr. Loftin, his students, and coworkers have been exploring the application of advanced software technologies such as artificial intelligence and interactive computer graphics to the development of training systems. Much of this work has been done in cooperation with the Software Technology Branch of the National Aeronautics and Space Administration's Lyndon B. Johnson Space Center in Houston, Texas. Completed and on-going projects include intelligent computer-aided training systems for astronauts, flight controllers, test engineers, and computer operators; an intelligent tutoring system for high school and college physics students; and a virtual environment used in preparing the flight team for the Hubble Space Telescope maintenance mission. Dr. Loftin has been the recipient of numerous awards, including the American Association of Artificial Intelligence Award for an Innovative Application of Artificial Intelligence, NASA's Space Act Award, and the NASA Public Service Medal. He is the author of more than ninety technical presentations and publications.

Patrick J. Kenney graduated from the University of Wisconsin-Oshkosh in 1984 with a B.S. in Physics and from the University of Houston-Clear Lake in December, 1994 with a M.S. in Instructional Technology. He has supported NASA at the Johnson Space Center since 1984, first as a space shuttle navigation flight controller, then in shuttle cargo/payload integration and flight software verification. Since 1989, Mr. Kenney has supported the Software Technology Branch of the NASA/JSC Information Systems Directorate in the areas of technology evaluation, expert systems and multimedia applications development, and virtual training environments. His work with virtual environments involves development and prototyping, incorporation of mission-specific procedures and training scenarios, and integration of existing simulations and intelligent computer-aided training technology with VEs. Mr. Kenney's masters thesis involved the instructional design and technical development of a VE for studying planetary motion.

Virtual Environments in Training: NASA's Hubble Space Telescope Mission

R. Bowen Loftin, Patrick J. Kenney, *et al.*
University of Houston and NASA/Johnson Space Center, Houston, TX

INTRODUCTION

A rapidly maturing technology first proposed in the 1960s [Hall, 1963; Sutherland, 1968; Vickers, 1970] now offers a novel, unique avenue for the delivery of experiential training to personnel in many disciplines. Usually described as "virtual reality" (or virtual environments or synthetic environments or virtual worlds or artificial reality), this technology can provide both visual and auditory information of such fidelity that the observer can "suspend disbelief" and accept that he or she is actually somewhere else [Chung, 1989]. Further, the technology also permits perceptual and tactile interaction with the synthetic environment, enabling the user to transcend the role of passive observer and actively participate in shaping events [Minsky, 1990].

Extensive research and development in virtual environment technology has been stimulated by its application in areas such as engineering design [Orr, 1989], architecture [Brooks, 1987], data visualization [Brooks, 1988; Fuchs, 1989], and teleoperation [McGreevy, 1991]. Brooks and his coworkers at the University of North Carolina at Chapel Hill [Batter, 1972; Ouh-Young, 1988] have extensively explored the visualization and tactile/force feedback aspects of virtual reality for use by chemists and biochemists (in viewing, assembling, and manipulating molecules), but they have not addressed training or educational applications of the technology. Some education-related projects currently underway are typically directed at providing students with the tools needed to construct their own virtual world and the ability to explore that world [Merickel, 1990; McCormick, 1991; Byrne, 1992; McCluskey, 1992], rather than developing a reality whose implicit structure is based on pedagogical principles. The work reported here joins a small group of efforts that have specifically examined the training efficacy of virtual environments [Regian, 1992; Knerr, 1993; Kozak, 1993]. A closely-related project is also investigating the use of virtual environments in science education [Loftin, 1993].

NASA'S VIRTUAL ENVIRONMENT TECHNOLOGY FOR TRAINING PROGRAM

The NASA/JSC Software Technology Branch (STB) has been exploring the application of Virtual Environment Technology to training since 1990. Simulations of elements of Space Station Freedom and Space Shuttle payloads (such as the IntelSat captured during STS-49) have been developed to test the technology's efficacy as a training tool and to identify specific research and development needs to improve training performance. During 1993 a major project, replicating all relevant repair and maintenance scenarios for the Hubble Space Telescope mission (STS-61), was completed. Over one hundred members of the flight control team were trained, beginning in September, 1993, with this system, providing an opportunity to demonstrate the potential of virtual environment technology in training. Related activities within the STB include: (1) an evaluation of tactile, force, and temperature feedback mechanisms to enhance training transfer; (2) the integration of virtual environments with Intelligent Computer-Aided Training (ICAT) technology; (3) the sharing of Virtual Environments over long distances for collective training and concurrent engineering (with NASA/Marshall Space Flight Center), (4) the development of software tools that support the rapid development and maintenance of virtual environments by training personnel; and (5) the development of a Virtual Physics Laboratory as an educational "spinoff" of this NASA activity.

THE HUBBLE SPACE TELESCOPE REPAIR AND MAINTENANCE MISSION

The Hubble Space Telescope

> When Lyman Spitzer first proposed a great, earth-orbiting telescope in 1946, the nuclear source of stars had been known for just six years. External galaxies and the expanding universe were about twenty years of age in the human consciousness. Pluto was seventeen and the Seyfert galaxies were three. Quasars, black holes, gravitational lenses, and detection of the Big Bang were still in the future—together with much of what constitutes our current

understanding of the solar system and the cosmos beyond it. [Brown, 1991]

Shortly after the launching of the Hubble Space Telescope (HST), in April, 1990, astronomers became aware that its optical system was flawed. A commission, chaired by Robert Brown and Holland Ford, was convened to investigate the problem and propose a solution. The report of this commission [Brown, 1991] became the blueprint for the HST repair and maintenance mission. The preparation and crew training for that mission became a major focus of NASA for over three years.

Mission Training in a Virtual Environment

The Hubble Space Telescope repair and maintenance mission (STS-61) was successfully completed in December, 1993. As a part of the training employed for that mission, over 100 flight controllers actively experienced immersive virtual environments, simulating the extravehicular activities (EVA) that were planned for the mission. This effort was apparently the first large-scale implementation of Virtual Environment Technology for training personnel for a "real" mission. It was the result of a cooperative effort between the NASA Johnson Space Center's Software Technology Branch (PT4), the Space Flight Training Division, and the Flight Director Office. The primary training goal was to familiarize flight controllers, engineers, and technicians—all of whom were members of the mission's ground-based support team—with the location, appearance, and operability of the different components on the HST, as well as the related maintenance components in the Space Shuttle payload bay. Hence, the overall strategy of this project was to utilize VE for training while exploring the potential of VE training applications for space-related activities.

Training large numbers of flight controllers for any given Space Shuttle mission can be difficult due to the limited availability of training facilities and personnel. Important elements of hands-on training are conducted in heavily-scheduled facilities that are expensive to operate. Training time allocated for the primary flight control team is at a premium. First priority to the training facilities and simulations is usually given to this team and the crew. Much less time is available for the support and planning teams to train.

Crew training largely centers around the Weightless Environment Training Facility (WETF) in which astronauts can manipulate replicas of hardware components in neutral buoyancy. This training facility is expensive to operate and requires lengthy periods for training even one crew member. Thus, the WETF is generally unavailable for training non-astronaut members of a mission flight team. Other facilities, such as the Shuttle Mission Simulator, the air-bearing floor, and various engineering and part-task simulators are also generally unavailable for extensive use by ground-based support personnel.

With the complexity and importance of the Hubble Space Telescope repair and maintenance mission in December of 1993, other options for training flight support personnel needed to be considered. It was hypothesized that personnel involved with flight control, payload operations, and EVA planning could benefit from training in the HST and Space Shuttle cargo bay hardware configurations, equipment operation, and astronaut EVA procedures to more effectively support this mission. In particular, this mission contained more EVA operations than any previous mission and required extensive interaction between the flight crew and ground-based personnel. John Muratore, one of the three mission Flight Directors, suggested that a VE be developed to allow flight team members to gain an accurate knowledge of the HST geometry and the procedural steps to be followed in accomplishing the planned repairs and maintenance.

Development Software and Approach

Characteristics of the scenarios (*e.g.*, models, environment layout, behaviors, feedback) were created using two NASA-developed software tools, an ANSI C compiler, and a computer-based audio application. The Solid Surface Modeler (SSM) program was used to develop the individual environment models, or objects. SSM is a 3-dimensional (3D) graphics development application for solid-shaded and wireframe geometric modeling. This software tool was originally created at the NASA Johnson Space Center specifically for building detailed 3D objects for animations and conceptual simulations. As such, it is well-suited for virtual environment model creation and was used extensively in this project. Beginning with elementary shapes (*i.e.*, primitives), the complexity of objects was increased by combining these or altering them with geometric manipulations. Object surfaces were defined as flat or smooth with SSM, as were the color of objects using an 8-bit color palette.

The Tree Display Manager (TDM) is a real-time graphics visualization tool used to create a hierarchical representation (*i.e.*, a relationship tree) of the 3D models created with SSM. This tool allows developers to give structure and organization to the virtual HST scenarios, and to control users' viewing perspectives. It was also used to define the mobility and constraints of objects in a given environment as well as characteristics such as light sources, multiple views, and "trails" of an object's motion. The TDM tool was also developed at the NASA Johnson Space Center.

A Silicon Graphics, Inc. proprietary software graphics library (GL) was used to render the virtual environments. Likewise, the device controller and six scenario applications were developed in ANSI C to interrogate the input devices' data and to code behaviors, characteristics, and interactions between objects and the user within the virtual environment.

To accommodate the use of audio feedback in the training, the SoundTool utility provided on Sun/Sparc workstations was used. This software has the capability of recording sound and storing the data as digital audio files, and playing these files over internal or external speaker systems. All of the sounds associated with the six HST scenarios (*i.e.*, object identification and status messages) were recorded with SoundTool. A database of sound message codes and sound file names was created for rapid, real-time identification. A conventional external speaker was used for sound projection so that both active and passive participants could hear the feedback.

The full HST servicing and repair mission virtual environment training system included graphical representations, or models, of the HST, the Space Shuttle cargo bay, and maintenance/replacement hardware necessary for the user to complete the major procedural steps associated with the planned EVA servicing activities. There were six EVA scenarios developed, comprising two virtual environment training modules. These scenarios coincided with the six primary EVAs scheduled for the actual mission and included:

1. Solar Array change-outs,
2. Rate Sensor Unit (RSU) change-outs,
3. Corrective Optics Space Telescope Axial Replacement (COSTAR),
4. Wide Field/Planetary Camera (WF/PC II) change-out,
5. Solar Array Drive Adapter Electronics (SADE) replacement, and
6. Magnetic Sensing System (MSS) - Magnetometer installation over original Magnetometers

The specific procedural steps associated with each of the EVA tasks were determined from the Extravehicular Activity Annex for the HST First Servicing Mission (NSTS 14009, Annex 11, Revision B) and from extensive reviews of video recordings of astronauts undergoing training in the WETF. By developing the environment objects and hardware in accordance with engineering drawings, accurate and realistic models of the real objects were provided. The actual detail and fidelity of these environment models was ultimately dictated by the intricacy of the corresponding procedures. The behaviors and operations of these objects were also accurately portrayed in the virtual environment as necessary for completion of the stated goals. Figure 1 is a typical scene from one of the scenarios.

VIRTUAL ENVIRONMENT TRAINING APPROACH

Training Objectives

Since the terminal goals of this system were to transfer knowledge of the HST hardware and EVA procedures, the intended approach to the training addressed both cognitive and psychomotor skills. Before a learner entered the virtual HST environment, he/she already had some basic knowledge of the hardware components and task procedures from previous experiences and document study. The virtual environment therefore provided learners with a three dimensional view of what they had previously seen only in two dimensional drawings or photographs. It was believed the users' cognitive skills associated with task procedures would be improved because of the learner's ability to visualize a particular hardware component or astronaut's orientation in space, and because the learner was expected to recall the correct procedural steps in the correct sequence to accomplish a particular task. Levels of guided learning were used with respect to the "selectability" of an object out of the proper procedural sequence and with the instructional aids available to the user. These components of the training system were based on limited Intelligent Computer-Aided Training capability that was integrated with the virtual environment [Loftin, 1991, 1994]. An important consideration in executing a particular procedural step has to do

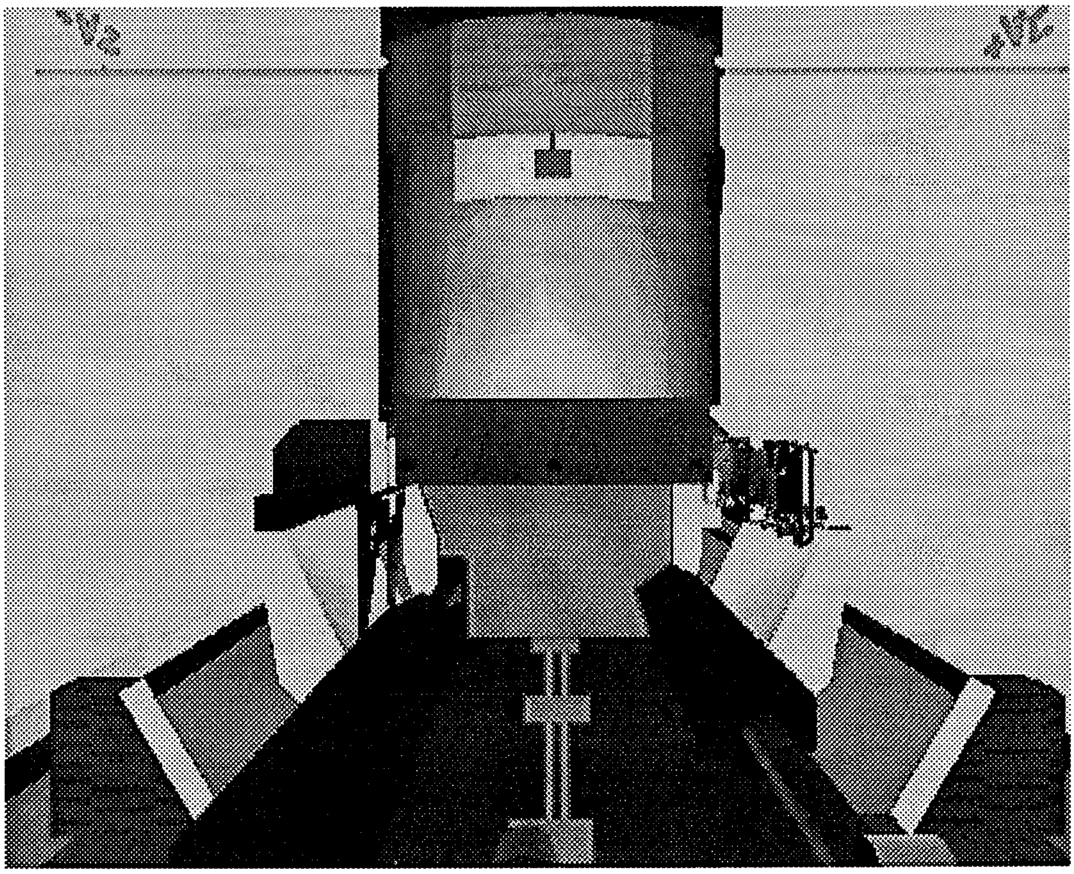


FIGURE 1. A view of the HST from the forward payload bay area. The axes and identifying letters shown were used for reference and orientation purposes.

with an astronaut's orientation to the hardware component. In these virtual training environments, the learner could maneuver with six degrees of freedom and experience this orientation first-hand to gain a better understanding of the psychomotor and visualization skills required of astronauts.

Description of Subjects

Users of the training system were 105 flight controllers who were actively immersed in the virtual environment using a head-mounted display system. Other members of the flight support team were able to observe using a monitor located near the subject. The trainees had varying levels of familiarity with the HST repair mission documented procedures and hardware components. All were highly motivated to learn the material and capable of mastering the user actions necessary for interaction in the virtual environment. It was assumed that the target group for this training had moderate to high learning and achievement capabilities, and excellent problem-solving skills.

In completing the virtual tasks associated with the mission's EVA procedures, it was anticipated the training system users would have several prerequisite skills and basic knowledge. In particular, this included: (1) knowing the correct sequence of EVA activities; (2) awareness of the position and orientation of the HST, fixtures, components, and astronauts; (3) identifying hardware components; and (4) specific tasks associated with the repair/maintenance of individual components (*e.g.*, bolts to be removed in sequence).

Post-Flight Subject Survey

From the beginning of September, 1993 until the first week of the space shuttle HST repair mission in December of 1993, 105 flight controllers were trained with the VE EVA scenarios. Several of these trainees were involved in more than one training session. After this VE training and the HST mission were completed, a questionnaire was developed and sent to all trainees to survey their impressions and comments on the HST VE training scenarios. The purposes of this questionnaire were threefold: (1) to study the effectiveness of this training in enhancing the flight controllers' performance during the HST mission, (2) to evaluate the training potential of VE technology, and (3) to assess some of the human factors issues and user-to-environment interface methodologies afforded by the training system. The survey feedback will be used to improve future VE training applications and to evolve virtual technology as an effective training medium by improving the content of virtual environments (*i.e.*, graphics and instruction), and users' interactions within them. In developing these survey questions, both instructional issues and technical aspects of the HST training scenarios were addressed. A copy of the questionnaire, as well as the high, low, and average of responses to each question can be obtained from the primary authors.

The survey contained four sections: (1) Personal Data, (2) Session Data, (3) Physical Data, and (4) Improvements and Suggestions. Within these sections, there were four types, or formats, of questions asked. The most frequently used was a ranking scale called a "Likert" scale. This type of scale was deemed to be the most appropriate in most cases as the purpose of this questionnaire was to assess more subjective concepts such as perceptions, attitudes, and self-evaluations. Numerical values corresponding to various adjectives describing users' experiences served to evaluate the sample and recommendations. The second most frequent type of question was a simple checklist, which was primarily used to collect data on the specific virtual environment hardware used and the training scenarios experienced. A third format was also a checklist type, but incorporated a ranking scale. Questions with this format pertained to the respondents' physical side-effects and severity. Finally, the fourth type of question gathered optional written responses and comments. These solicited rationale for responses to many of the individual questions and suggestions for improvement of virtual reality training applications.

RESULTS

Survey Returns

Of the 105 surveys mailed to trainees, 38 completed forms were returned. Whereas completion rates on mail questionnaires are typically low—with figures of 40 or 50 percent considered good—the 38-questionnaire response on this mailing was typical of voluntary return rates. As nearly all surveys returned were completed in their entirety, the results should form a solid basis for assessing and improving virtual reality training applications. On occasion, some questions were not answered. Therefore, the compiled results referenced in this paper are based on the number of answers received for each question. Similarly, written comments to questions were somewhat sporadic, with some questions receiving many far-ranging remarks while others had none.

Survey Results

An important finding concerned the overall effectiveness rating of training. Users graded overall effectiveness at "slightly over effective" or 4.08 (out of a range of 0 - 5). Figure 2 shows the lowest, highest, and average rating for the overall case as well as that for each of the six scenarios. Comments showed that users found that visualizing activities enhanced understanding. Comments from subjects also noted that efficiency in training delivery was also increased, corresponding to the ability to compress EVA (Extravehicular Activity) time from hours into minutes. For example, a five hour EVA would take thirty-five minutes in HST/VR.

At an average of 3.75 (out of a range of 0 - 5), users reported "just below significant" knowledge gains from the HST/VR training experience. The ability to visualize tasks (and positions of various items in the shuttle cargo bay and on the HST) had a positive impact on user's comprehension of activities and objects. Apparently, the user's prior knowledge influenced ratings somewhat negatively where experts reported not learning as much as novices. However, this should not be mistaken as a negative outcome.

The audio and instructional aids (*i.e.*, audio messages and visual cues) of HST/VR received high marks of 4.2 (out of a range of 0 - 5) for their enhancement of the "immersiveness" of the experience. The majority of comments corresponding to this question were all very positive and ranged from just "Fun!" to being ". . . the neatest training tool . . ." that individual had ever used. Respondents reported that VR training or replication of EVA procedures would have been beneficial early on in the flight planning activities. Also, the HST/VR system could have been used as a prototyping tool for task reconfiguration or troubleshooting.

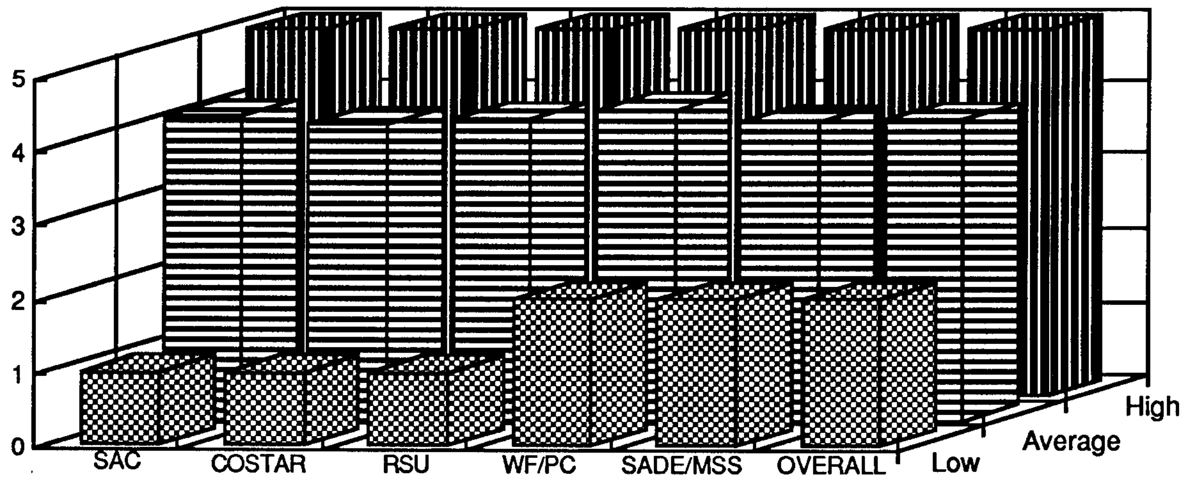


FIGURE 2. Subject ratings of performance enhancement due to the use of the HST Virtual Environment Training System. High and low ratings, in addition to the average rating are shown for five scenarios and for the overall training experience.

An interesting trend from the questionnaire dealt with physical side-effects such as nausea, oculomotor problems, and disorientation resulting in mild cases of simulator-sickness. Apparently, compared with current and past simulator sickness studies [Kennedy, 1992], HST/VR trainees reported relatively lower rates of the various side-effects elaborated in the survey. Figure 3 shows the rate of occurrence of these side-effects by type.

Based on reports and comments received from flight controllers during the actual STS-61 HST mission operations, the HST/VR development team received compliments, during and after the mission, on its realistic modeling of the shuttle cargo bay and HST. Several flight controllers compared down-linked video data with what they had experienced in VR. The result seemed to be empathy and instant recognition of those objects associated with the HST/VR project scenario and the real mission.

CONCLUSIONS

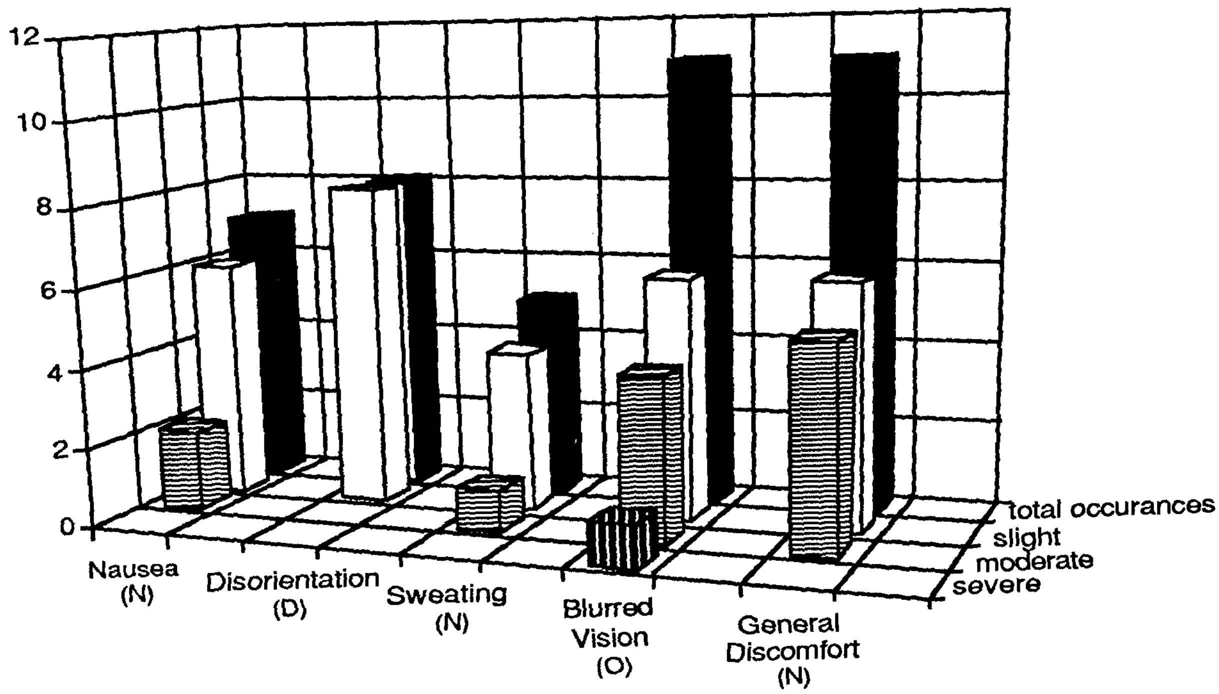
As the results above demonstrate, members of the flight team judged, on the average, that the use of a virtual environment for training had a positive effect on their job performance during the HST repair and maintenance mission. The discomfort experienced by many of the participants did not pose a serious problem with training transfer but should be studied further. A number of approaches that can reduce this discomfort or better manage it have been proposed and will be explored in future work [Kennedy, 1992]. The positive experiences reported here have broadened and deepened the interest of NASA in the use of Virtual EnvironmentTechnology as a training tool. Perhaps just as important is the opportunity that VEs afford for the training of personnel who currently receive little or no experiential preparation for their assigned task.

FUTURE WORK

One main activity that continued after the successful completion of both the HST/VR training project and STS-61 mission was the completion of a third, albeit lower priority, training module. This will accommodate anticipated servicing and repair tasks for the future HST oriented missions.

More ICAT functionality will be incorporated into future VR applications development. Such issues as lesson planning and management could benefit from the delivery of such VR based training systems.

As VR hardware technologies improve, efficacy in training and simulation will likely result. The search for advanced Head Mounted Displays will provide improved resolution and field-of-views to enhance visualization and, therefore, immersion capabilities. Improved haptic (*e.g.*, tactile) sensing capabilities will also expand functionality of training application to various tasks. Faster computing hardware will also serve to improve the capacity for overall VR application development efforts like the HST/VR project.



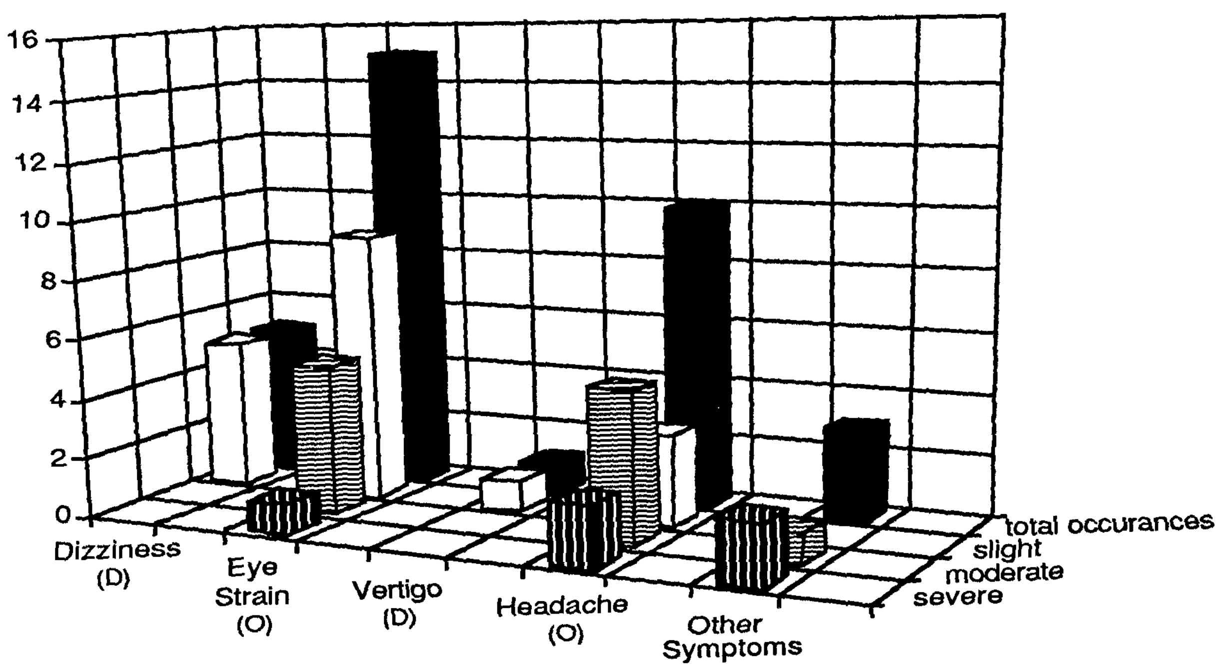


FIGURE 3. These two charts display the occurence of simulator sickness symptoms among the subjects. Symptoms are classified as nausea (N), disorientation (D), and oculomotor (O) after Kennedy, 1992.

REFERENCES

[Batter, 1972] J. J. Batter and F. P. Brooks, Jr., "GROPE-1," *IFIPS Proceedings 71*, 759.

[Brooks, 1987] F. P. Brooks, Jr., "Walk Through—A Dynamic Graphics System for Simulating Virtual Buildings," *Computer Graphics 21* (1), 3 (January, 1987).

[Brooks, 1988] F. P. Brooks, "Grasping Reality through Illusion—Interactive Graphics Serving Science," in *Proceedings of ACM SIGCHI*, Washington, DC, 1-11, 1988.

[Brown, 1991] R. A. Brown and H. C. Ford, "Report of the HST Strategy Panel: A Strategy for Recovery," (Baltimore, MD: Space Telescope Science Institute).

[Byrne, 1992] C. M. Byrne, personal communication, Human Interface Technology Laboratory, University of Washington, Seattle, WA, October 26, 1992.

[Chung, 1989] J. C. Chung, M. R. Harris, F. P. Brooks, Jr., H. Fuchs, M. T. Kelley, J. Hughes, M. Ouh-Young, C. Cheung, R. L. Holloway, and M. Pique, "Exploring Virtual Worlds with Head-Mounted Displays," in *Proceedings of the SPIE Conference on Three-Dimensional Visualization and Display Technologies*, Los Angeles, January 18-20, 1990, 42-52.

[Fuchs, 1989] H. Fuchs, M. Levoy, and S. M. Pizer, "Interactive Visualization of 3-D Medical Data," *IEEE Computer*, 46-50 (August, 1989).

[Hall, 1963] M. R. Hall and J. W. Miller, "Head-Mounted Electro-Ocular Display: A New Display Concept for Specialized Environments," *Aerospace Medicine*, 316-318 (April, 1963).

[Kennedy, 1992] R. S. Kennedy, N. E. Lane, M. G. Lilienthal, K. S. Berbaum, and L. J. Hettinger, "Profile Analysis of Simulator Sickness Symptoms: Application to Virtual Environment Systems". *Presence, 1*, pp. 295-301.

[Knerr, 1993] B. W. Knerr, D. R. Lampton, J. P. Bliss, J. M. Moshell, and B. S. Blau, "Assessing Human Performance in Virtual Worlds," Proceedings of the 1993 Conference on Intelligent Computer-Aided Training and Virtual Environment Technology (Houston, TX: NASA/Johnson Space Center), Vol. II, p. 270.

[Kozak, 1993] J. J. Kozak, P. A. Hancock, E. J. Arthur, and S. T. Chrysler, "Transfer of Training from Virtual Reality," *Ergonomics 36*, pp. 777-784.

[Loftin, 1992] R. B. Loftin and R. T. Savely, "Advanced Training Systems for the Next Decade and Beyond," *Proceedings of the AIAA Space Programs and Technologies Conference* (Washington, DC: American Institute for Aeronautics and Astronautics) AIAA Paper 92-1626.

[Loftin, 1993] R. B. Loftin, R. B. Loftin, M. Engelberg, and R. Benedetti, "Applying Virtual Reality in Education: A Prototypical Virtual Physics Laboratory," *Proceedings of the IEEE 1993 Symposium on Research Frontiers in Virtual Reality* (Los Alamitos, CA: IEEE Computer Society Press), pp. 67-74.

[Loftin, 1994] R. B. Loftin, L. Wang, P. Baffes, and G. Hua, "General Purpose Architecture for Intelligent Computer-Aided Training," U.S. Patent Number 5,311,422, awarded May, 1994.

[McCluskey, 1992] J. McCluskey, "A Primer on Virtual Reality," *THE Journal 20* (5), 56-59, December, 1992.

[McCormick, 1991] A. McCormick, panel presentation on work at Nueva Media Lab, SRI International Conference on Virtual Worlds: Real Challenges, June 18, 1991.

[McGreevy, 1991] M. W. McGreevy, "Virtual Reality and Planetary Exploration," in *Proceedings of the 29th AAS Goddard Memorial Symposium*, Washington, DC, March, 1991.

[Merickel, 1990] M. L. Merickel, "The Creative Technologies Project: Will Training in 2D/3D Graphics Enhance Kids' Cognitive Skills?," *THE Journal 18* (5), 55-58 (December, 1990).

[Minsky, 1990] M. Minsky, M. Ouh-Young, O. Steele, F. P. Brooks, Jr., and M. Behensky, "Feeling and Seeing: Issues in Force Display," in *Proceedings of Symposium of 3-D Interactive Graphics*, Snowbird, Utah, March, 1990.

[Orr, 1989] J. N. Orr, "Exotic CAD," *Computer Graphics World 12* (7), 88-89 (July, 1989).

[Ouh-Young, 1988] M. Ouh-Young, M. Pique, J. Hughes, N. Srinivasan, and F. P. Brooks, Jr., "Using a Manipulator for Force Display in

Molecular Docking," IEEE publication CH2555-1/88, 1824-1829.

[Regian, 1992] J. W. Regian, W. L. Shebilske, and J. M. Monk, "Virtual Reality: An Instructional Medium for Visual-Spatial Tasks," *J. Communication 42*, pp. 136-149.

[Sutherland, 1968] I. E. Sutherland, "Head-Mounted Three-Dimensional Display," in *Proceedings of the Fall Joint Computing Conference 33*, 757-764 (1968).

[Vickers, 1970] D. Vickers, "Head-Mounted Display Terminal," in *Proceedings of the IEEE Int. Computer Group Conference,* 102-109, 1970.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the willingness of members of the HST repair and maintenance mission flight team to endure some of the difficulties associated with current virtual environment hardware and to complete a lengthy post-flight survey. The work of Sean McRae in data analysis and reporting was invaluable to this project. This work was supported, in part, by the NASA Office of Space Systems and by NASA Grant NAG 9-713.

TRAINING DEVELOPMENT AND DELIVERY SUBCOMMITTEE

Chair
Stephen Stankiewicz, Evans and Sutherland

Deputy Chair
Cheval Force, Loral Federal Systems

Members
Al Galbavy, STRICOM
Arthur Marubbio, TASC
Bill Jorgensen, TSTC, Inc.
Brenda Hoskins, Lockheed Aeronautical Systems
Dan Hagan, AAI Corporation
Dr. Lois Wilson, Loral Aerospace
Dr. Michael Spector,USAF
Jackie Townsend, USAF
John Buckley, HQ TRADOC
Katherine Golas, Southwest Research Institute
Ken Larrabee, NAWCTSD
Larry Graham, Grumman Melbourne Systems
LTC Don Poe, USAF
Margot Moore, HQ TRADOC
Randy Buss, Unisys
Tom Moberg, CAE-Link Binghamton

Section 3
Table of Contents
Training, Development and Delivery Papers

A METHODOLOGY FOR SELECTION OF TRAINING TO APPLY COMPUTER-BASED INSTRUCTION

LTC David W. Raes
ARPA SIMITAR PROJECT
Camp Dodge Support Team

ABSTRACT

The purpose of this paper is to present a methodology for identification of training that computer-based instruction can be applied to in order to maximize training effectiveness. The process described is being developed as part of the Advanced Research Projects Agency SIMITAR (Simulation in Training for Advanced Readiness) initiative.

The project objective was to develop prototype individual and leader training approach for Forward Support Battalions of the Army National Guard. The scope of the problem included fifty two separate Military Occupational Specialties, as well as six different career fields for officers. The goal is to achieve 200-300% improvement in training effectiveness in the available time.

A "Lane Training" approach was used to isolate hard to train, high payoff, tasks to be developed using Computer-based Instruction. Lanes are developed using a top down analysis of missions, critical collective sub-tasks, as well as supporting leader and individual tasks. This pyramidal approach allows subject matter experts to filter critical tasks from the myriad of knowledges, skills, and abilities which seemingly carry the same level of importance.

Although this methodology is being applied to a military organization, the lane training approach can be applied to any entity. It effectively focuses organizational training objectives by breaking down priority organizational goals and the critical management and individual knowledges, skills, and abilities that are essential to organizational success.

The results of this project include: a methodology for focusing training priorities from the organizational mission to every leader/manager, and soldier/employee; a methodology for selecting high payoff tasks for CBI development; and a Training Management System to track individual and organizational status of training.

ABOUT THE AUTHOR

Lieutenant Colonel David W. Raes is a Battalion Commander in the Iowa Army National Guard. He is on a short tour with the Advanced Research Projects Agency as a Subject Matter Expert for Combat Service Support Training Development in the ARPA SIMITAR Project. His address is Iowa National Guard, ATTN: DOIM, 7700 NW Beaver Drive, Johnston, Iowa 50131-1902, ph: 515-252-4428.

A METHODOLOGY FOR SELECTION OF TRAINING TO APPLY COMPUTER-BASED INSTRUCTION

LTC David W. Raes
ARPA SIMITAR Project
Camp Dodge Support Team

INTRODUCTION

The Advanced Research Projects Agency's SIMITAR (Simulation in Training for Advanced Readiness) Program was initiated by Congressionally added funds in FY92 "... to apply advanced technology to the training of National Guard Roundout Brigades.", in response to training readiness challenges identified during the Desert Storm mobilization.

The SIMITAR program is a six year project involving two National Guard Roundout Brigades as experimental units and culminates in their mobilization and movement to the National Training Center for rotations using experimental technologies and training strategies.

The objective of SIMITAR is to achieve training readiness levels 200-300% higher than those observed in the mobilization for Desert Storm. This is to be accomplished through the development and extensive use of leading edge information technologies and radical new training strategies.

One element of the SIMITAR program is focused on the Combat Service Support training strategy. This effort is being accomplished at Camp Dodge, Johnston, Iowa, the Headquarters of the Iowa National Guard.

To highlight the training challenge facing the Combat Service Support Commander, the Divisional Forward Support Battalion (FSB) is used as an example. The FSB commander has 52 different military occupational specialties (MOS's). Each MOS represents several hundred skills that are applicable to each soldier. In addition, current trends indicate consolidation of related MOS's which will increase the total number of tasks for each soldier to master. Considering the fact that TRADOC Schools training only a small percentage of the total number of tasks, the training challenge faced by the CSS Commander is significant.

This paper outlines a methodology for identifying critical tasks that are of the highest priority for training and applies an objective method for selecting from those critical tasks ones to apply computer-based instruction (CBI) in order to maximize training effectiveness. Although applied to a military organization, this methodology could be adapted to any organization.

BACKGROUND

Current Environment

Most companies in today's environment are recognizing the need for increased skills training among employees. Workers are expected not only to do more but to have a better understanding of the Corporation and the critical cross-functional requirements. Likewise, in the military it is becoming more critical for soldiers to understand the responsibilities for, and the linkage between, the collective mission-essential tasks of the unit as a whole, and the individual tasks that support them.

The problem with task identification in today's environment is not lack of information or lack of training materials. In fact, the problem may be magnified by the mountain of training materials and general information available. There seems to be something wrong with the picture of a soldier preparing to deploy "down range" adorned with a Kevelar helmet, load-bearing equipment, individual weapon and two large brief cases bulging with essential reference manuals. However, it is a picture frequently seen. The fact is that the Army promises to increase the volume of information that any one soldier is responsible for as a result of consolidation of occupational specialties. Consolidation of work requirements is also a trend in the civilian sector. Access to virtual mountains of information is seemingly critical to job performance.

The key to effective organizational and individual training is to identify work elements that are truly critical to success. In other words, define the job. What's important? What can I get by without? We must filter out non-critical tasks in order to focus training resources on mission-essential tasks. In reality, the truly critical skills are generally discovered by the employee/soldier on the job. They learn what skills are essential by using a trail and error process. After a significant amount of "time in the trenches" the core critical functions are identified and learned. Even more significant is the fact that the lessons are learned all over again by each new employee or soldier, and by the same trial and error process. That is inefficient. It also runs the risk of having some employees or soldiers in critical positions without having been successful at the trial and error learning.

METHOD

Defining the Mission

The Army provides a series of reference documents that generally do an adequate job of defining unit missions, mission essential tasks, and subordinate collective tasks. Some Mission Training Plans (MTP's) also identify related soldier and leader tasks. The problem with MTP's is that they are too compartmental in their design. The result is that leaders and soldiers learn their own specific role but do not understand their relationship with other activities, sections, or units. Individual to collective matrices identify tasks for one echelon and do not address the critical linkages between echelons. Lessons learned at the National Training Center indicate that the execution of critical missions tends to break down at the coordination points, the linkages between echelons on the battlefield.

The problem of developing a compartmentalized view of the mission is being addressed in another SIMITAR element at the Army Research Institute, Presideo of Monterey, California. BDM Federal is developing a series of documents outlining Critical Combat Functions (CCF's). Critical Combat Functions are defined as "The integration of related players and tasks that represent a source of combat power. The synchronization of critical combat functions provides maneuver commanders at any echelon with a definable outcome that materially affects the battle." Each CCF document contains a flowchart depicting the relationships between echelons, key participants by task, a critical task list, and lessons learned at the Combat Training Centers, relevant to the specific CCF.

Defining the organization's primary missions is an important first step in prioritizing training. As mentioned above, the unit's Mission Training Plan is the source document for accomplishing this step in the Army. For each mission the Critical Combat Functions (CCF's) outline the relationship between key people and units at all echelons. Lane Training expands the view of a particular collective mission by including training focus on the actions required at connecting echelons.

What is Lane Training?

The term "lane training" is a bit confusing in that it suggests many different things to different people. Lane training originated in Infantry units in the Army. Typical Infantry training lanes required squads, platoons, or even companies to negotiate over a designated piece of terrain in order to accomplish their mission. They started in an assembly area, crossed a line of departure, navigated over a predetermined axis or route to an objective, thus the term lane training.

Today the lane training strategy is used by all types of Army units. When observed in other units the term lane training is a bit of a misnomer because in many cases the lane is one static location. For example, a lane for a Maintenance unit might be to repair a diesel power plant/pack. This lane requires a maintenance bay or field static location and all of the resources necessary to perform the tasks for the team participating in the training. The term lane doesn't seem to fit here but the lane training strategy works as well with this type of unit as it does with the Infantry unit.

The definition of lane training is a technique for training Company level and subordinate platoons and sections on a series of related leader and individual tasks that are critical to the successful execution of priority mission-essential tasks. Lanes are developed using a top down analysis of missions, critical collective sub-tasks, as well as supporting leader and individual tasks. This pyramidal approach allows subject matter experts to filter critical tasks from the myriad of knowledges, skills and abilities which seemingly carry the same level of importance. Lanes also reinforce the critical linkages between echelons and provide the vehicle to train on the techniques and procedures that contribute to mission accomplishment at all levels.

Lane Development

Each mission-essential task for each company of the Forward Support Battalion are broken down into subordinate collective lanes. Each lane is then dissected to identify related leader and individual tasks. This process aligns tasks to missions, but individual and leader tasks need another filter applied to reduce further the total volume of tasks to be trained. The filter that was applied to reduce the total number of tasks for each lane was subjective; the criterion was to select only those tasks that if not performed adequately would impact adversely on the successful completion of the collective mission of the lane. This process provides leaders and soldiers with a realistic number of tasks to attain and maintain proficiency. In addition, it maintains focus on the commander's mission essential tasks to attain and maintain unit competencies.

The lane training approach applies a development structure to ensure that each completed activity will be of maximum usefulness even if time prevents further training. If there is time for just one more training activity, what should it be?

Implementation of Lane Training

The preferred method of training in the Army is hands-on. If training can be resourced, it should be conducted on the actual equipment and as close to the same conditions as would be expected in combat as possible. Many units find it very difficult to resource hands-on lanes on a regular basis during Inactive-duty training at their local armories. Most units can resource lanes during their annual training cycle but training on critical collective tasks once a year is not adequate to develop or sustain task proficiency.

What generally happens in units when critical tasks cannot be trained due to a resource constraint is to schedule training on other, easier to resource, tasks. The problem that this creates is not evident at first glance because effective training may be occurring. However, valuable training time is being spent on lower priority tasks at the expense of training on more critical tasks that support priority lanes.

It is this problem that our project is addressing by applying computer-based training technology. By developing CBI for tasks that are difficult to train due to some resource constraint, commanders and managers can ensure that training time is spent on high priority skill development.

Selection of Tasks for CBI Development

An objective approach was developed to consider tasks for application of computer-based instruction. Many studies have been done on media selection for training development and basically all of them indicate that no one media is the total answer to training needs. The assumption applied to this project was that if hands-on training was difficult for the unit to conduct at the Armory then computer-based technology would be applied to provide a method to train on the task. It was also assumed that computer-based instruction would not replace hands-on training but would prepare soldiers and leaders by providing preparatory training on critical tasks so that when hands-on training could be conducted, soldiers would progress much faster to achieve proficiency. The hands-on training opportunity would be a validation of the effectiveness of computer-based instruction. This phase of the project has yet to be measured. However, a plan is being developed to capture this data for analysis.

Quality computer-based instruction is not cheap to produce. In order to get the most for the budget, a quantitative approach was used to select tasks for CBI development. Decision matrices were formulated to apply a numerical value to all of the priority individual and leader tasks that were critical to each lane. Figure 1 illustrates the decision criteria or states of nature that were applied in the decision matrix for evaluation. These criteria are a combination of the general advantages of computer-based instruction which could be applicable to any training material and specific conditions that are present in the military to inhibit hands-on training. Table 1 is a sample of how the decision matrices were laid out.

The decision matrices also provide for the application of weights to the criteria. In our situation it was decided that the most important aspect to consider was whether the tasks were difficult to train in the armory environment. A weight of 2 was applied to this criteria. All other criteria were given a weight of 1. This same objective approach could be used by any organization to determine tasks for CBI development. The specific criteria could be modified as well as the relative importance of each criterion.

By applying a quantitative method, a priority list is produced. This allows for CBI development to progress based on availability of funds in a training priority sequence.

RESULTS

The initial SIMITAR objective was to develop prototype individual and leader training for Forward Support Battalions of the National Guard. The scope of the problem included addressing 52 Military Occupational Specialties with literally hundreds of required knowledges, skills, and abilities each. The overall goal of the project is to achieve 200-300% improvement in training effectiveness in the time available.

A lane training strategy has been used to prioritize tasks for training. This is accomplished by dissecting organizational goals and objectives and reducing them down to the core level individual and leader tasks that are critical to mission accomplishment. The key to the lane training strategy is attaining individual and collective proficiency on critical tasks first. Although this seems logical and straightforward, very few organizations achieve and maintain this level of training focus. One problem is that some tasks critical to the lanes are difficult or impossible to train at the Armory due to some resource constraint.

The application of computer-based instruction - applying an objective measurement to determine the highest payoff tasks - has been an effective and efficient use of development resources.

DISCUSSION

The goal of developing a methodology for selection of training to apply computer-based instruction produced two separate procedures to accomplish the objective.

The first step was to determine a method for prioritization of tasks for training. Lane training, although not invented here, seemed to be the best training strategy for isolating the highest payoff individual and leader tasks for the target units in the Forward Support Battalions.

Once the priority individual and leader tasks for the unit were identified, the second step applied an objective method of selecting the tasks for CBI development. This procedure produces a prioritized list of high payoff tasks for CBI development.

Although the results of the procedures outlined here have not yet been validated, the initial reactions from customers are very positive. It certainly is better than a haphazard approach to the application of development resources to computer-based instruction.

CONCLUSIONS

Multi-media hardware and software developers are making great strides in improving the development tools. Although quality computer-based instruction requires significant development time, efficient production of CBI is improving on a daily basis. The power of interactive computer-based instruction makes it a highly potential area for meeting training needs. It is imperative that development resources are utilized in an effective and efficient manner. This project presents an approach to this process.

ACKNOWLEDGMENTS

The author would like to thank Dr. William A. Deterline for his professional assistance in the preparation of this paper.

Figure 1
DECISION CRITERIA FOR CBI DEVELOPMENT
(States of Nature)

SPECIFIC CRITERIA

- **COMPLEXITY OF TASK** - Difficulty of training the task due to the technical skill required.
- **LOW DENSITY CRITICAL TASK** - Tasks that are critical to the mission of the unit but involve only a few individuals.
- **EQUIPMENT SHORTAGE** - Lack of equipment on which to train.
- **SUPPORT EQUIPMENT** - Lack of special tools or test equipment to conduct training.
- **FACILITIES** - Insufficient or inadequate facilities or training areas.
- **LACK OF TRAINERS** - Qualified trainers are not available.
- **TIME INTENSIVE** - An extensive amount of time is required to set up and conduct training.
- **SAFETY** - Students can learn potentially dangerous tasks without risk.

GENERAL CRITERIA

- **REDUCED LEARNING TIME** - Studies indicate that interactive learning is as much as 50% more efficient than traditional training techniques.
- **REDUCED COST** - By comparison with CBI, hands-on lane training is very resource intensive.
- **INSTRUCTIONAL CONSISTENCY** - Hands -on training conducted by each first line supervisor can be inconsistent in quality and accuracy.
- **PRIVACY** - Peer pressure can adversely effect learning. Each student can work on a task until mastery without fear of being criticized by peers.
- **MASTERY OF LEARNING** - A structured program of instruction provides a map for logical learning.
- **INCREASED RETENTION** - Interaction between the student and CBI provides increased retention over time.
- **INCREASED MOTIVATION** - Interactive learning has proven to be very motivational for the student.
- **INCREASED ACCESS** - Quality training can be available to the student at the time and place convenient to each student.

TABLE 1

DECISION MATRIX FOR SELECTION OF CBI TASKS

UNIT: LANE: TASK:	COMPLEXITY OF TASK	LOW DENSITY CRITICAL TASK	EQUIPMENT SHORTAGE	SUPPORT EQUIPMENT	FACILITIES	LACK OF TRAINER	TIME INTENSIVE	REDUCED COST	REDUCED LEARNING TIME	TRAINING INCONSISTENCY	PRIVACY	MASTERY OF LEARNING	INCREASED RETENTION	INCREASED SAFETY	INCREASED MOTIVATION	INCREASED ACCESS	
WEIGHT TASK	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	WEIGHTED VALUE

AUTOMATED AUTHORING: SOME PRELIMINARY RESULTS

William J. Walsh
Mei Technology Corporation
San Antonio, Texas

ABSTRACT

This paper reports preliminary results of research into automated authoring currently being conducted by Mei Technology for Armstrong Laboratory. The preliminary results reported here are based on an internal try-out of the eXperimental Advanced Instructional Design Advisor (XAIDA). XAIDA is an expert system which automatically generates computer-based training from system information provided by a subject matter expert. XAIDA is not a finished product. The system is undergoing formative evaluation at this time. Currently, a small group of experienced instructional designers and some novices are using XAIDA to develop courseware. These internal tryouts will result in modifications to the system prior to more extensive tests with actual Air Force users (both authors and students). A brief description of the research foundations of the program are followed by an outline of the authoring process and student presentation. Some XAIDA features which affect authors and students are also described. Experienced and novice authors' reactions to the system from internal try-outs are reported, including problems encountered, time to author, authors preferences and lessons learned.

ABOUT THE AUTHOR

William J. Walsh has provided technical direction for Mei Technology's Training Technology Division for the past five years. Prior to that he was involved in the design and development of training systems and researching training technology issues for over 15 years including prototyping experimental systems for Air Education and Training Command under the Training Technology Applications Program. He has worked on and managed programs involving implementation of various training technologies, including computer-based training, multimedia applications, intelligent computer-assisted training, simulations of maintenance and troubleshooting, and distance learning among others. Recently his concentration has been on technological applications to reduce instructional development time and increase instructional quality through automated authoring.

AUTOMATED AUTHORING: SOME PRELIMINARY RESULTS

William J. Walsh
Mei Technology Corporation
San Antonio, Texas

INTRODUCTION

Experts agree that the quality of computer-based instruction can vary greatly depending on a number of variables such as instructional design skills of the author and tools available to create interesting designs. Training managers frequently cringe at the up-front investment required to author computer-based training materials no matter what the quality. Armstrong Laboratory has been conducting research into the simplification of the instructional design process for several years. One of their initiatives is development of the eXperimental Advanced Instructional Design Advisor (XAIDA).

Growing Demand for Technology-based Technical Training. Computer-based instruction is a powerful tool when placed in the hands of a skilled instructional developer. However, the availability of skilled instructional developers in the Air Force is limited. Traditionally, the Air Force relies on a few instructional developers to assist many subject matter experts to develop training. If tools were available to assist the instructional designer, i.e., to *clone* the expert instructional developer, more high quality computer-based instruction could be produced faster and cheaper than with current practices.

The laboratory's work on XAIDA is on the verge of providing a breakthrough in the area of instructional design tools. XAIDA's generative power, i.e., its ability to create new instruction from the kind of information provided by an instructionally naive SME, is a step in the right direction, answering both the quality and cost questions associated with computer-based instruction. This paper reports preliminary results of an *internal* try-out of XAIDA by several novice and experienced instructional designers. Results include authoring time for various types of lessons, estimates of instructional quality, lessons learned and a brief description of the kind of courseware produced.

Computer-Based Training Problems

There are numerous problems associated with computer-based training. Instructional design and development for computer-based training are repetitive, time-consuming activities traditionally associated with high cost. Good instructional design is crucial for quality courseware, yet the Air Force has few of these experts. Production of consistent quality courseware has been a difficult factor to measure (Goodyear, 1993).

Today, Air Force technical training consists of intensive school training followed up with extensive on-the-job training. Most technical courses rely on older technologies and are time and labor intensive to develop, maintain and deliver. When technical training students graduate they are only partially trained. In the future, it is the goal of the Air Force to produce mission ready graduates by providing more timeliness and balance to technical training, by making use of new technologies and by rapidly prototyping interactive courseware. Armstrong Laboratory's goal is to support the use of new technologies cost-effectively to improve the overall quality of instruction. In order to achieve that goal, the laboratory is researching automated performance support systems for instructional design.

The Concept of Automated Authoring

The industry has been dealing with automation of the courseware authoring process for years (Merrill, 1985). While early computer-based training was programmer intensive and required extensive scripting to produce text and graphics screens, more recent innovations in authoring, e.g., Authorware, IconAuthor[1], etc. are highly visually-oriented and are designed to be used by non-programmers. Newer authoring

---

[1] Authorware is a registered trademark of Macromedia, Inc. IconAuthor is a registered trademark of Aimtech Corp.

packages also use screen metaphors such as slide shows, linked screens, card stacks, etc. (Burger, 1994; Park, 1994). However, none of these approaches to simplify authoring through automation focus on a critical issue in authoring computer based courseware, i.e., providing instructional design assistance to novice authors so they can produce instructionally sound lessons.

There are several approaches to automating instructional design (Wasson, 1993, Halff, 1993). Two approaches are: 1) develop an authoring system that provides tools and automated processes for authors to create instruction, or 2) develop an authoring system which produces instruction from a representation of the subject matter? Armstrong Laboratory has already designed and developed a system using the first approach, the Guided Approach to Instructional Design Advising (GAIDA).[2] This system is also commonly called *Gagne's* Instructional Design Advisor because it is based on his ideas about automating the instructional design process. This system was developed by Gagne and several other laboratory scientists while he was in residence. GAIDA presents advice to authors on the nine events of instruction (Gagne, 1985). Authors can switch back and forth from a description of each event to an example from a lesson which demonstrates how the event could be portrayed. While GAIDA is classified as automated and does contain limited interaction, it does not contain what a *student model* which allows it to tailor instructional design advice to the specific content of a user's lesson or to other needs of the user. Regian and Shute (in press) conclude that "consistent guidance on automated instruction is not available," ... systems like GAIDA "provide useful information about *what* needs to be done in developing instruction, they tell very little about *how* to do it."

While GAIDA meets some immediate needs of Armstrong Laboratory and Air Force users for an on-line performance support system for novice instructional developers, XAIDA is aimed at providing the next generation of intelligent performance support tools for interactive courseware development. The following sections provide a brief description of XAIDA's origin, what it looks like now, characteristics of lessons currently generated by XAIDA, some indication of authoring efficiency, and plans for improvement of XAIDA.

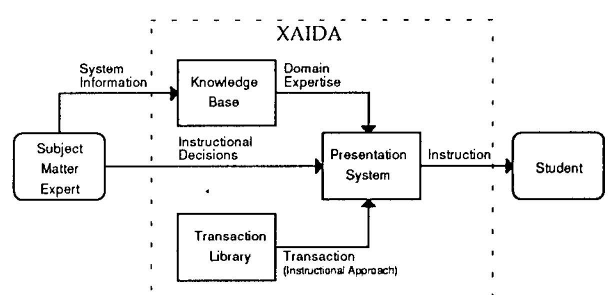
[2] Copyright Armstrong Laboratory, 1992.

XAIDA

Conceptual Foundation

XAIDA is a transaction-based, generative, fully automated authoring system. XAIDA utilizes Merrill's Second Generation Instructional Design Theory (Merrill, 1992, 1989, 1993) to automatically create interactive courseware from domain knowledge provided by a subject matter expert. An overview of how XAIDA works can be seen in figure 1. Essentially, a subject matter expert provides domain information to XAIDA regarding a specific aircraft system and XAIDA builds a knowledge base from this information. Some instructional decisions are made by the instructor such as the kind of lesson which will be presented to the student. Based on this input XAIDA then assigns a transaction to the lesson which determines how it will be presented to the student.

Figure 1. XAIDA Overview



While AIDA[3] is conceived as being able to provide instructional design advice and generate instruction for more than one domain, XAIDA currently focuses on maintenance training. Several basic assumptions were made early in the program which have influenced the development of AIDA. The assumptions were

[3] AIDA should be considered the general term for the overall research program concerning instructional design advising. The prototype version of the software is called XAIDA since it is still experimental and undergoing formative evaluation. For the complete history of this project see Hickey, Spector and Muraida (1992).

formulated into a list of guidelines for system development. Besides focusing on maintenance training, perhaps the most important assumption or guideline is that AIDA is not bound by the Air Force ISD model. As a result, XAIDA makes use of transaction shells loosely based on Merrill's Second Generation Instructional Design Theory.

The AIDA concept was the outgrowth of an idea of Dr. Scott Newcomb, chief of the Instructional Design branch and Dr. J. Michael Spector, a 1988 summer faculty researcher, now senior scientist in the branch. The AIDA program called upon a group of world renown psychologists[4] to formulate its basic tenets and approach. These psychologists were assisted by a panel of senior military advisors from several Air Force and other service training and research organizations. This group worked hard trying to reach agreement on how the AIDA program should achieve the goal of automating instructional design.

Figure 2. Course Structure

Course List	Course Outline
001: Course List	002: C-141 Oxygen System
002: C-141 Oxygen System	003: Components
005: Pre-Use Inspection of LOX	004: Operation

XAIDA Structure

XAIDA consists of four transaction shells: identify, interpret, execute and troubleshoot. These shells roughly correspond to the kinds of objectives involved in maintenance training. *Identify* is concerned with declarative knowledge such as component parts, nomenclature and location of components. *Interpret* is concerned with how systems operate, their inputs, outputs, rules and states. *Execute* is concerned with procedural knowledge, i.e., how to perform certain activities associated with maintenance such as removing a nosewheel tire, or replacing an oxygen regulator, etc. Troubleshoot is, as its name implies, concerned with fault isolation or troubleshooting. Three of these shells (identify, interpret and execute) have been implemented. While much research has already gone into the troubleshoot shell, it is not ready for implementation at this time. Work with troubleshooting will not begin until the first three shells have undergone further testing.

What XAIDA Looks Like Now[5]

Authoring Lessons. The authoring process in XAIDA routinely begins with the instructor setting up a curriculum outline. XAIDA uses a simple tree structure to depict the curriculum with a course having at least one and potentially several lessons (see figure 2). Each lesson is further broken down into at least one and usually several topics[6] (see figure 3). XAIDA's knowledge engineering begins to be noticeable at this point. When describing the various topics which make up an individual lesson, the instructor either describes the components of the system for an identify shell, the inputs, outputs and states of a system for an interpret lesson, or the procedures associated with operating or maintaining components of the system for an execute lesson. System elements described for one type of lesson can also be used for other types, e.g., the nomenclature and location of

Figure 3. Subject Matter

Process List	Process Outline
001: Process List	002: C-141 Ox. Sys. Components
002: C-141 Ox. Sys. Components	003: Crew Ox. System
	007: Pilot Regulator
	008: Co-Pilot Regulator
	009: Nav. Regulator
	010: Eng. Regulator
	004: Troop Ox. System
	005: Converter Pallet
	006: Regulator Panel

---

4 The research advisors consisted of Drs. Robert M. Gagne, Henry M. Halff, M. David Merrill, Martha C. Polson, Robert D. Tennyson, Harold F. O'Neil, Charles Reigeluth and Douglas M. Towne.

5 The version being discussed in the rest of this paper is XAIDA Version 2.6, September 1993.

6 Topic is a generic term which covers several XAIDA structural elements such as processes, activities, or properties.

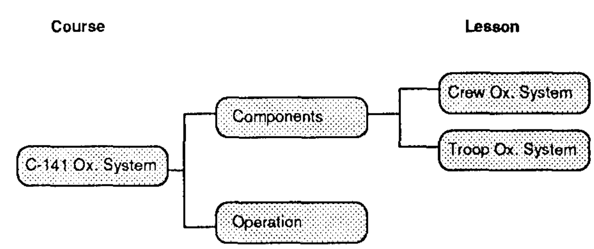
components described by a subject matter expert for an identify lesson can also be used in developing an interpret shell for the same avionics system with some additional amplifications.

Once the basic structure of the curriculum has been entered into XAIDA, resources can be added to any element of the course structure, i.e., course, lesson or topic level. Currently, resources can take the form of text, audio (*.WAV* files), or video (*.AVI* files). XAIDA does not create resources, rather, it calls upon existing Windows[7] packages to create or modify resources. In addition to these resource types, XAIDA interfaces with two simulation packages (PowerSim 1.1 and Rapid 2.0)[8] which can also be used.

After the required information has been provided to XAIDA and resources have been created or modified for the lesson(s) to be presented the instructor selects a transaction. Transactions are normally applied at the lesson level. In other words, once the instructor knows what content he/she wants taught in a lesson, namely the parts of a system, or how it works, or what procedure must be used to fix a component, all that remains is to edit the parameters of the lesson(s) at the Course/Unit Editor. Editing these parameters allows the instructor to set some of the automatic switches used in XAIDA. For example, one of the choices an instructor will make is the transaction type to be used in the lesson, i.e., identify, interpret or execute. The instructor also selects other course control elements from a list of items which are then combined by XAIDA to generate a lesson.

Presenting a Lesson. When taking a lesson in XAIDA, a student is first presented with the course outline developed by the instructor. From this hierarchical outline the student can choose which course(s) and which lesson(s) within a course to take (see figure 4). While this is the way XAIDA operates now, there has been some consideration given to allowing for either student control of lesson choice (as is now the case), or system assignment of lessons based on predetermined factors set by the instructor. These features may be implemented in later versions of XAIDA. Once a student has selected a course and lesson, the presentation begins. Currently, most lessons begin with an optional overview. The overview consists of a part-to-whole or bottom-up presentation of each lesson topic. After the overview, XAIDA goes through the lesson content provided by the subject matter expert in accordance with the predetermined algorithm for the selected transaction. For the *identify* shell, this consists of presentation of an explanation of each system component or part as described in the lesson with text, graphics, audio and/or video resources. Identify can also present the steps of a procedure without going into a simulation of the maintenance activity. The description is followed by an optional review of any portion of the presentation. The next lesson segment makes use of the same resources to ask the student to identify the name of system components or locate them on a graphic.

Figure 4. Lesson Hierarchy



For the *interpret* shell, the lesson presentation is slightly different. An interpret lesson consists of an optional overview of the system just like identify, however, the interpret lesson presents the rules, states and properties of the system being studied rather than just explaining component parts and their location. After presenting a description of the system, the interpret lesson allows the student to interact with the rules which govern the system by changing inputs, outputs or system states and observing the effect this has on the system. Finally, the student is presented questions regarding the system states and rules which he/she must answer before going on to another lesson.

For the *execute* shell, the lesson is different from both identify and interpret. An execute

[7] Windows is a registered trademark of Microsoft Corp.

[8] PowerSim copyrighted © by ModellData AS. Rapid is a product of Emultek.

lesson can be: 1) a demonstration of the steps of a maintenance procedure, 2) guided practice, i.e., prompts from XAIDA as to the step/procedure to be performed, or 3) unguided practice -- similar to testing the student's knowledge of the procedure. For example, a student can assemble or disassemble a component or tell how the specific steps of an activity are performed. The student must select tools and the correct actions to be performed with a tool on a designated component.

RESULTS

XAIDA is currently being tested with Air Force subject matter experts, Armstrong Laboratory scientists, Mei Technology instructional developers, and outside consultants. Each of these groups has a different focus when looking at XAIDA. Figure 5 shows the evaluation matrix used for XAIDA. Some of the preliminary results are discussed in the following sections; anecdotal data, collected from these try-outs, are reported here. Most summative data regarding XAIDA will be collected at Sheppard AFB in 1995, although some limited preliminary data will be available in Fall of 1994.

Figure 5. XAIDA Evaluation Matrix

	Product	Process
Authors	Authoring System • Using the tool to develop CBT • Complete • Accurate • Interface	• Can authors use tool • Time • Efficiency • Complies with Air Force methods
Studentst	Courseware • Students learn from lessons • Complete • Accurate • Interface	• Student results • Effectiveness • Efficiency • Complies with Air Force methods

Sheppard AFB Try-outs

Informal testing of XAIDA began almost immediately after a prototype system was developed. These first tests were conducted with two instructors from the Air Force representative of the target population. The very brief try-outs usually consisted of a few days on site at Sheppard AFB with the subjects guided through the authoring process by the XAIDA system programmer. Although this one-on-one attention was necessary at first, the instructors, who participated in several sessions, soon became very familiar with XAIDA and how to author with it. Once trained on how XAIDA operates the Air Force instructors were able to create all kinds of lessons within a few hours.

The Sheppard AFB instructors had a very positive reaction to XAIDA. They felt that authoring was easier and less time consuming with XAIDA compared to commercial packages they were using. Using the commercial authoring package they reported it took about 30 days to develop a 30 minute lesson. Using XAIDA one instructor developed a lesson on *Hydraulic System Pressurization* in about 3 hours. Their students' reactions to the lessons was equally positive. The instructors reported that the students who took a lesson on *KC-10 Flight Controls* were really impressed with what they were able to do on their own and retained quite a bit of knowledge after using XAIDA.

Internal Try-outs

A group of professional instructional developers used XAIDA for several weeks to evaluate various elements of the software. The evaluation plan included evaluation of both product and processes were to be evaluated and the variety of target audiences who will be using XAIDA. The various components evaluated are depicted in figure 5. Instructional designers and psychologists were tasked with looking at XAIDA in terms of how the authoring system performed and how well a student could learn from the lessons produced by the system. In other words, the evaluators focused on all aspects of XAIDA.

Authoring. At first, our instructional designers had more difficulty using XAIDA than the subjects at Sheppard AFB. This is attributed to two factors. First, the Sheppard AFB instructors had direct access and support from the programmer who was developing XAIDA. He was able to advise them firsthand about what worked and what didn't, and the authoring procedure(s) to follow in order to provide XAIDA's knowledge engineering component with the necessary information. Second, a user's manual had not yet been written, therefore, the instructional designers had little to rely on except trial and error or their ability to remember XAIDA functions that had been demonstrated to them. In spite of this, sample lessons were developed by all the experts and all but one of the novices within a few days.[9]

While everyone involved in the internal try-outs was able to develop an identify lesson within a very short time, more problems were experienced in attempting to produce an interpret lesson. Both experienced and inexperienced instructional developers had trouble visualizing the kind of information structure needed by XAIDA to formulate rules for the interpret shell. These problems triggered development of the first version of a user's manual. After some struggling, authors were able to produce interpret lessons of various lengths for several objectives. While developing these test lessons some fundamental questions were asked by the instructional designers about whether interpret, as it was originally designed, corresponded with how maintenance personnel viewed their job. Identify lessons appeared to be satisfactory in teaching system nomenclature and location, but interpret lessons tend to be more like limited simulations of the system rather than building true mental models. The instructional designers felt that XAIDA should remain closely aligned with the maintenance process, i.e., declarative knowledge about components, mental models of systems, procedural knowledge of maintenance activities and troubleshooting strategies for finding and diagnosing faults.

Data from these early try-outs were used to modify XAIDA's authoring and delivery components. The instructional developers made recommendations about the authoring interface, transaction shells, feedback to the author, and several other features. Some of the more experienced authors felt that XAIDA should actively provide prompts to the author about such things as what information needs to be filled in next and information the system needs prior to being able to present a lesson. Terminology presented a major point of irritation for everyone. For example, authors liked the way the course structure is developed in XAIDA, but they were not happy with terms XAIDA used to refer to lessons, topics, objectives, or other typical course elements. One of the things they felt might be confusing to Air Force instructors is the use of terms like transaction, process, properties, focus, portrayal, and other such terms. In their opinion, XAIDA terminology should mirror more closely the instructors' own maintenance training jargon rather than the abstract reasoning processes underlying XAIDA.

The major authoring problem for the novices was recognizing what had to be done next in order to produce a fully functional lesson. At first novices felt that XAIDA had no predetermined sequence of steps which had to be followed. They had trouble remembering all the steps and the sequence of events which led to development of a student-ready lesson. Once they internalized the XAIDA authoring procedure they were faced with other problems. When developing interpret lessons the novices weren't able to comprehend what XAIDA's *property class library* was or how it fit into the interpret shell. The property class library defines the behavior characteristics of various system components for XAIDA. Eventually, the novices learned how to build elements for the library by assigning values to the states of components. Once they were able to construct these XAIDA building blocks constructing rules which governed the system was a relatively easy task for them.

Preconceived Ideas about Authoring. The more experienced instructional developers were initially disturbed by how XAIDA authored. In their past authoring experience they usually developed some design plan for how they wanted the lesson to be taught, then went about implementing it with the authoring tools at hand. This is not how XAIDA operates. While XAIDA starts with the author providing a course outline, it needs system information about each component which comprises a curriculum element to produce a lesson. In other words, while the experienced authors wanted to work on course design, XAIDA forced them into decomposing knowledge about particular systems. The traditional work of the instructional designer, i.e., creating a course design, was being done by XAIDA. Like it or not, the authors were being forced into the role of subject matter expert! This is not so surprising since the goal of the Air Force is to improve the quality of instruction while decreasing reliance on such skilled specialists.

9 Even my first lesson (an identify shell) was developed in about 7 hours. About an hour of development time was wasted because an XAIDA file was missing from my computer.

Authoring Time. One of the key questions to be answered eventually by this research is whether XAIDA reduces the amount of time needed to author computer-based training materials. From some of the anecdotal data presented here, indications are that authors are able to create lessons in far less time using XAIDA than with commercial authoring products. While the data produced thus far indicates a reduction in authoring time, formal evaluation of XAIDA authoring time has not yet been conducted. The first formal tests of authoring time will consist of collecting data on how long it takes subjects[10] to create an identify lesson, given sample lessons for the same objective, an authoring manual, and limited support from an experienced author, e.g., a brief walk through of how the software operates and stop-gap help if all else fails. Eventually, time to create lessons in XAIDA will be compared against development times for the same lessons in commercial products being used by the Air Force, e.g., Sabre, Mandarin[11] or other commercial authoring products.

Informal data from the novice instructional developers provide some indication of XAIDA's authoring time. After overcoming the initial hurdle learning XAIDA's terminology and authoring procedures (between eighteen to twenty four hours spread over two weeks), one novice author was able to develop five identify lessons on the *Universal Aerial Refueling Receptacle Slipway Installation (UARRSI)*, the *UARRSI Door/ Slipway, Refueling Receptacle, Aerial Refueling Manifold, and Isolation Valves* within a single day. The same author took slightly longer (about 18 hours) to develop two components of an interpret lesson on the refueling system. Authoring time does not include graphics development for either of these lessons.[12] The lessons developed comprise a significant portion of a two hour class (of a total fourteen hour unit) devoted to fuel systems in the Airlift Aerospace Maintenance Apprentice (C-141) course. The most proficient novice XAIDA author is developing identify lessons in one and one half hours not including graphics, video or audio resources. The same novice author can also develop more complex interpret lessons in about four hours.

Student Perspective. The experienced instructional designers also had strong opinions about how XAIDA presented a lesson. In general, lesson presentation in XAIDA consists of an overview of the system, presentation of the system information, some student freeplay or review of the learning materials and answering questions automatically generated by XAIDA. The exact details of lesson presentation differ based on the transaction shell being used. Some presentation characteristics which were viewed as problems include: the overview is too structured and system oriented, lack of objectives, information windows overlay graphics, and question types are too restrictive. Some of these problems are related to the fact that Version 2.6 of XAIDA is still prototype software which has not been compiled into a fully functioning software package. In other words, the programmer was waiting for several substantive changes to be identified through the try-outs before making other minor adjustments needed to ensure program elegance. Typical of this kind of change is locating XAIDA's information windows in a specific screen area so that there is no interference with viewing the text and associated graphics at the same time. Several changes have been identified for programming into later versions of XAIDA, including correction of perceived problems such as the lack of statement of learning outcomes or objectives prior to beginning the lesson and the confining nature of the overview presentation where the student has to go through each element of the system before being able to do anything else.

All of the current changes identified for XAIDA's presentation system are based on preferences of the authors based on their past experience and foundation in sound learning theory rather than data from student try-outs. While some very small groups of students at Sheppard AFB have used the XAIDA programs

[10] Novice authors for this experiment are from college level psychology and statistics classes.

[11] Sabre is being used by Keesler AFB. Mandarin was developed by Marconi Simulation and is being used by the Air Force in the Advanced Training System.

[12] Graphics development time could vary depending on whether bitmaps or video are being used. We make it a point to try to segregate graphics development time whenever possible. XAIDA has no impact on time to develop graphics.

for training, testing with the target student population is not scheduled until mid to late 1995.

Measures of Instructional Quality

One of the goals in evaluating XAIDA is to determine the quality of instruction which can be generated by an automated instructional design system. Two things are necessary in order to accomplish this. First, an objective measure of courseware quality must be established and accepted by the Air Force as valid, and second we must determine some way of assessing XAIDA lessons in terms of where they fit on the scale of instructional quality. As mentioned earlier, one method of determining lesson quality which will eventually be used on this project is to compare XAIDA lessons with those of other authoring systems. This subjective method of evaluating authoring efficiency will be used on some of the XAIDA generated lessons which will be tested in side-by-side comparisons. This type of evaluation produces data of limited use. A better test will be to compare both types of lessons against a more objective indicator of instructional quality. The laboratory and Air Education and Training Command have been queried as to whether the Air Force has such a measure. Indications are that various local yardsticks exist, however, very few organizations are willing to hold their own measures up to scrutiny for use on this or other such projects. Perhaps, such measures of instructional quality can be extracted from the guidelines produced for the AIDA program (Hickey, 1992) or the new Air Force instructional systems development regulations, e.g., AF Handbook 36-2235.

CONCLUSIONS

Significance of Automated Authoring. Based on the preliminary findings reported here, automation of the instructional design process has far-reaching implications for the development of interactive courseware. The experienced authors using XAIDA reported that they felt a little frustrated using the system because they weren't exercising the strength of their skill repertoire, i.e., creating original instructional designs. The generation of interactive courseware by means of automated instructional design systems renders the creative author into a subject matter expert, or at least a knowledge engineer. While the reaction from experienced instructional designers appears to cast a negative light on automated systems such as XAIDA, the reactions of novice developers was quite the opposite. No such complaints were heard from the novices. Novices who used XAIDA did not have any preconceived ideas about how they should be designing instruction. In fact, once they learned the system and terminology their authoring proficiency was about equal to that of the experts. Development time was reduced significantly for both groups. As has been pointed out, the only factor which remains to be assessed is the quality of the interactive courseware generated by automated systems. If XAIDA's courseware is judged to be of satisfactory quality, the courseware produced by the novices should be equal to the experts, since all courseware produced by XAIDA is basically the same.

Implications for Research. The research plan for XAIDA calls for further formative evaluation consisting of additional internal testing with instructional designers, testing in a typical Air Force setting with experienced and novice authors, evaluation of the interactive courseware produced by XAIDA with students from the actual target population, and comparison of the courseware with lessons on the same material produced with other authoring packages and against an objective quality standard. Furthermore, testing of XAIDA will examine the application of user adaptive guidance in systems such as XAIDA. Ideally, an authoring system should be able to adapt its guidance to the level of expertise of the developer. One way to accomplish this is to vary the level of prescriptiveness, another is to vary the extent of explanations given to the developer. In order to design such an adaptive authoring system, one needs information about how guidance should change as expertise increases. The use of media such as graphics, video, and audio in addition to text offers the potential of great educational benefit. Some content matter (e.g., how to interpret an X-ray or an echocardiogram) may be inherently better suited to presentation in a non-text format. Also, some students learn better in non-text formats. However, some research has shown that, in other cases, a traditional text-only presentation leads to better learning than combining text with other modalities. Further research will investigate when and how multimedia features should be combined in courseware in order to optimize learning and how a system like XAIDA can generate specifically tailored courseware to accommodate these learning styles.

While these tasks are scheduled on the research agenda, several additional areas still remain to be investigated. Among potential research topics are: the ability of automated courseware generation systems like XAIDA to make use of electronically stored maintenance information to generate courseware; incorporation of intelligent tutoring features such as a student model into systems like XAIDA; and, the ability of XAIDA and other such authoring systems to easily incorporate other instructional design strategies.

REFERENCES

Burger, J. (1994). New Directions in Authoring. *New Media*, 4 (5), 44-48.

Gagne, R.D. (1985). *The Conditions of Learning*, 4th ed. New York: Holt, Rinehart, and Winston.

Goodyear, P. (1993). Foundations for Courseware Engineering. In *Automating Instructional Design, Development, and Delivery*, ed. R. D. Tennyson. Berlin: Springer-Verlag.

Halff, H.M. (1993). Prospects for Automating Instructional Design. In *Automating Instructional Design: Concepts and Issues*, ed. J.M. Spector, M.C. Polson & D.J. Muraida. Englewood Cliffs, NJ: Educational Technology Publications.

Hickey, A.E., Spector, J.M. and Muraida, D.J. (1992). *Design Specifications for the Advanced Instructional Design Advisor (AIDA)*, AL-TR-1991-0085. Brooks AFB, TX: Armstrong Laboratory.

Merrill, M.D. (1985). Where is the Authoring in Authoring Systems? *Journal of Computer-Based Instruction*, 12 (4), 90-96.

Merrill, M.D. and Li, Z. 1989. An Instructional Design Expert System. *Journal of Computer-Based Instruction*, 15 (3), 95-101.

Merrill, M.D. (1992). A General Theory of Instructional Design. In *Design Specifications for the Advanced Instructional Design Advisor (AIDA)*, ed. A.E. Hickey, J.M. Spector & D.J. Muraida. AL-TR-1991-0085-Vol-1. Armstrong Laboratory, Brooks AFB, TX.

Merrill, M.D. (1993). An Integrated Model for Automating Instructional Design and Delivery. In *Automating Instructional Design: Concepts and Issues*, ed. J.M. Spector, M.C. Polson & D.J. Muraida. Englewood Cliffs, NJ: Educational Technology Publications.

Park, W.T. (1994). Future Trends in Authoring. *New Media*, 4 (5), 52-54.

Regian, J. W. and Shute, V.J. (In press). Basic Research on the Pedagogy of Automated Instruction. In *Use of Computer Models for Explication, Analysis and Experiential Learning*, ed. T. De Jong, H. Spada, & D. Towne. Berlin: Springer-Verlag.

U.S. Air Force, Headquarters. *Information for Designers of Instructional Systems: Interactive Courseware (ICW) Design, Development, and Management Guide.* AF Handbook 36-2235, Volume 5. Washington, D.C.: Department of the Air Force.

Wasson, B. (1993). Automating the Development of Intelligent Learning Environments: A Perspective on Implementation Issues. In *Automating Instructional Design, Development, and Delivery*, ed. R. D. Tennyson. Berlin: Springer-Verlag.

Acknowledgments - The research reported here was funded under Contract F41624-93-D-5002 by the U. S. Air Force Armstrong Laboratory, Human Resources Directorate, Brooks AFB, Texas.

APPLICATION OF TRAINING ANALYSIS AND DESIGN TOOLS

Dr Michael Reakes
Westland System Assessment Limited
Yeovil, United Kingdom

Mr William Carpenter
Westland Helicopters Limited
Yeovil, United Kingdom

ABSTRACT

This paper describes the application of a software tool in the training analysis and curriculum design of a large multi-national helicopter project. Requirements for the tool are outlined, and the selection process is described. Commercially-available tools are reviewed, and the application of such a tool at Westland Helicopter's Customer Training School is described.

When a new helicopter is developed, airframe manufacturers consider training requirements as part of the overall logistics plan. A training curriculum and an integrated suite of training media (typically ranging from Computer-Based Training to Simulators) are specified and procured. A systematic approach to the training analysis, design, development, implementation and evaluation is required to provide an objective, auditable record of the decision making process, and to allow project controls to be applied. The analysis of maintenance and operator tasks, selection of tasks for training, development and sequencing of learning objectives, and the specification of appropriate training media, are some of the key steps in creating a successful and cost-effective training system.

Software tools are very effective in supporting training analysis and design by guiding analysts through the required decision making processes, allowing them to make quicker and more consistent training decisions. The tools automatically create auditable and traceable records of the decision processes. Logistics Support Analysis Records can be imported as the starting point for the maintenance analysis, helping to integrate the training system with the evolving aircraft design. Training data can be easily and quickly stored, retrieved, shared and exchanged - resulting in a reduction in duplicated data. Configuration control facilities allow changes to be tracked as the aircraft design evolves. Use of a common software tool, data dictionary and database structure allows interchange of computer-readable training data amongst geographically distant organisations.

ABOUT THE AUTHORS

Dr Michael Reakes is a Senior Consultant at Westland System Assessment Limited, located in Yeovil, United Kingdom (UK). Dr Reakes has more than 15 years experience in the design, development and implementation of training systems in the USA and UK. He has led the selection and implementation of training analysis and development tools at Westland System Assessment.

Mr William Carpenter is the EH101 Training Team Leader at Westland Helicopters Customer Training School. Mr Carpenter is responsible for the design and implementation of an integrated suite of training equipment and courses to support the EH101 helicopter.

APPLICATION OF TRAINING ANALYSIS AND DESIGN TOOLS

Dr Michael Reakes
Westland System Assessment Limited
Yeovil, United Kingdom

Mr William Carpenter
Westland Helicopters Limited
Yeovil, United Kingdom

INTRODUCTION

When a new aircraft is developed (the EH101 helicopter, for example) airframe manufacturers such as Westland Helicopters consider training requirements as part of the overall logistics plan. A training curriculum and an integrated suite of training media are specified, procured, and used to support training courses for helicopter operation and maintenance. The training media and courses must be in place before the delivery of the first helicopter. A typical suite of training media for helicopter maintenance and operator training is depicted in Figure 1, and ranges from Computer-Based Training, through Part-Task Trainers to Simulators.

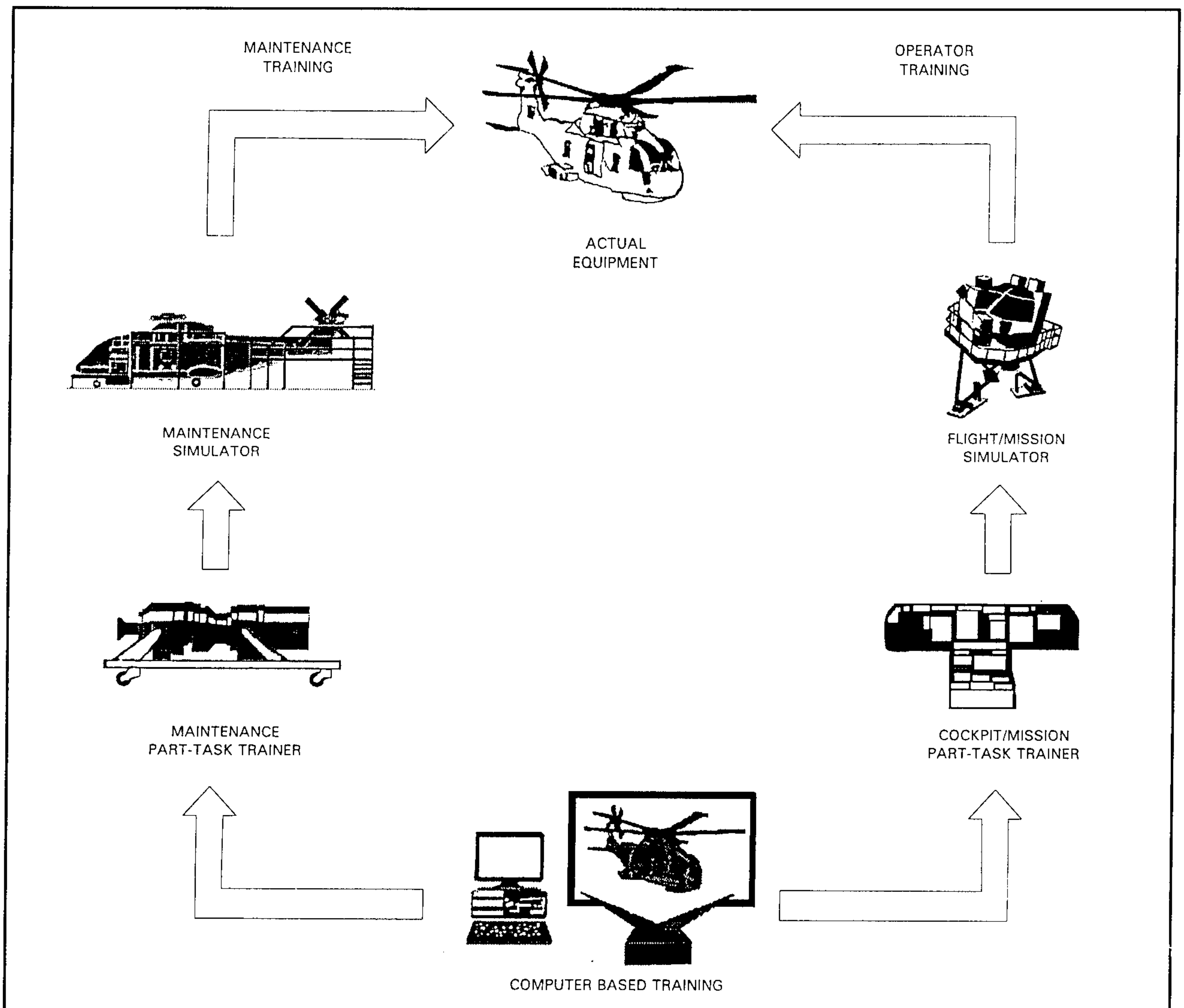


Figure 1 - Typical Training Media for a New Helicopter

Such a suite of training devices is a major expense, and it is essential that the specifications and learning objectives for each training device are systematically established and justified, so that the resulting training is cost-effective.

THE ISD PROCESS

The Instructional Systems Development (ISD) process [1][2] provides a closed-loop iterative approach to the Analysis, Design, Development, Implementation, and Control of large instructional systems (Figure 2).

ISD is a flexible step-by-step process for planning and developing instructional systems which ensures personnel are taught the knowledge, skills, and attitudes essential for successful job performance in a cost-effective way. ISD is also known as the Systems Approach to Training [3][4].

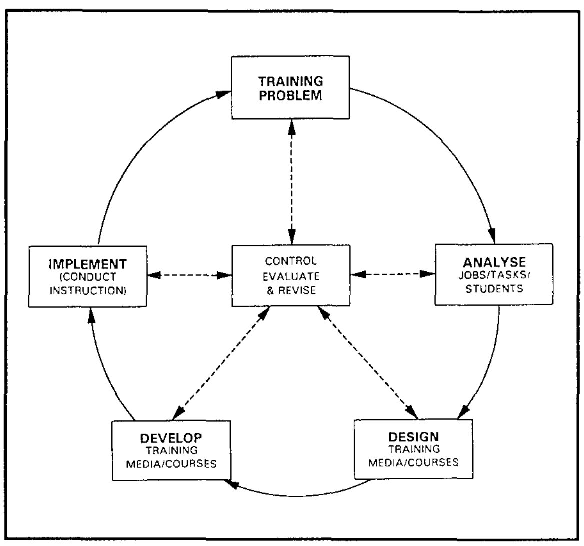


Figure 2 - Overview of the ISD Process

The following key steps from the Analysis, Design and Development stages of the ISD process are employed when developing helicopter training system for aircrew and maintainers [5]:

- Define Aircraft Systems
- Analyze Job Categories
- Analyze Tasks & Performance Measures
- Define Target Trainee Population
- Select Tasks for Training
- Select Instructional Setting
- Develop Learning Objectives & Performance Measures
- Select Training Media
- Develop Training Device Specifications
- Develop Courses

THE NEED FOR A COMPUTERISED TOOL

The ISD process is led by a Project Manager who oversees a team of Training Analysts and Subject Matter Experts (SMEs). In the past, the task selection process, media selection process, and other key ISD steps, were implemented manually, by following a set of written guidelines. The results were documented on paper. The process worked, but tended to suffer from the following disadvantages:

Labour Intensive - The detailed implementation of each of the key steps was a very labour intensive process.

Rework - Considerable effort was involved each time the training analysis data was evaluated and revised as part of the ISD process. Additionally, considerable rework effort was necessary each time the aircraft system design changed.

Inconsistencies - Training decisions tended to be inconsistent from analyst to analyst, and from project to project. Each analyst tended to have a slightly different interpretation of the written guidelines.

Duplication - Paper documents could not easily be shared while they were being developed. This led to duplication and redundant effort amongst the team of analysts.

Use of LSARs - Logistic Support Analysis is the iterative process that regularly updates the new weapon system's design and supportability information through all phases of development. LSARs contain design and logistics information for the aircraft or weapons system. The detailed steps involved in maintenance procedures for aircraft systems are often contained within pre-existing Logistic Support Analysis Records (LSARs) [6], and form the basis for maintenance Tasks and Subtasks.

With a manual process, LSA data was available to the analysts only as printed reports, and records were re-entered. This was inefficient, subject to error, and created rework when the LSAR records were updated.

Audit Trail - It was difficult to search and find information quickly amongst paper records to check why any given decision was made.

Process Training - When a new project started, and when a new analyst joined the team, considerable effort was necessary to train the analysts in how to apply the written guidelines.

Geographically Separate Analysis Teams - The EH101 helicopter is a cooperative venture between Westland in the UK and Agusta in Italy. Training analysis and design is conducted separately in the two countries, and merged to form a whole. A mechanism was needed to allow the two geographically-separate analysis teams to follow consistent procedures, and to merge the results into a single training design database.

Variants - There was a need to take the "core" training analysis for the basic aircraft, and merge it with extra training analysis for each new variant (e.g. mission systems such as sonar and radar). This was time consuming with the manual process.

Supply Machine Readable Data - Increasingly, customers are demanding the results of the training analysis to be supplied to them in a structured machine readable form, as part of the contract for the helicopter. This was the case with the variants of the EH101 proposed for Canada. The training analysis for the basic vehicle was to be handed over in machine readable format to the Canadian Prime Contractor, who would add the training analysis for the mission systems.

REQUIREMENTS SPECIFICATION

For these reasons, the authors set up a project to specify and select a computerised training analysis tool for use on the EH101 project, and future similar projects. To form the basis of the selection process we prepared a Requirements Specification. The following is a synopsis of the requirements:

Training Analysis Methodologies - The tool must be able to implement the custom methodology and terminology selected for use on the project [5]; this included both maintenance and aircrew training analysis. The tool must also be capable of meeting the standard ISD processes (such as MIL-STD-1379D [2], and A-P9-000 [3]) used by potential customers throughout the world. The ability of the tool to comply with the analysis process and terminology outlined in Canadian Forces Publication A-P9-000 [3] was an important requirement.

Aircraft System Definition - The tool must allow the training analyst to define the aircraft Systems and Subsystems using an agreed coding system such as AECMA 1000D [7].

Job/Task Analysis - The tool must be capable of performing job and task analyses for operating and maintaining a modern aircraft. The tool must assist training analysts by providing facilities for verifying the completeness of data by, for example, searching for inconsistencies or incomplete data.

Job Analysis - The tool must allow the definition of job categories. For aircrew, job categories may include the Pilot, Copilot, Flight Engineer, and Mission Specialists (e.g. Sonar Operators). For maintenance personnel, jobs are defined by civilian and military trade specialities and levels, such as Electrical/Avionics, Mechanical, and Weapons.

Task Analysis - The tool must allow the definition of Tasks and Subtasks for each job category. For flight operations, this is normally done by phase of flight (Preflight, Engine Start, Taxi, etc). For maintenance, the breakdown is for the Conduct and Management of Flight Servicing, Scheduled Maintenance, and Unscheduled Maintenance (i.e. diagnosis & repair). Further breakdown is by aircraft system and subsystem. The tool must support the breakdown of Tasks and Subtasks in terms of the supporting Knowledge, Skills and Attitudes.

Input of LSA Data - The tool must quickly and easily be able to import and utilise LSA data [6] for potential use in a maintenance task analysis.

Iteration - The task analysis is conducted at various levels, depending upon the availability of detailed task data. Early analysis will use conceptual tasks, which will be gradually replaced with the specific details from LSA data as the analysis and design progresses. This enables a progressive definition of the training system, which allows early definition of long-lead training media requirements, such as maintenance or flight simulators. The tool must facilitate this iterative process.

Define Target Trainee Population - The tool should allow the analyst to define the characteristics of the target trainee population. The definition is normally in terms of the Knowledge, Skills and Attitudes displayed by the average prospective trainee prior to the commencement of the training.

Select Tasks for Training - This step in the ISD process ensures that instruction is provided for all important tasks, but that resources are not wasted on unimportant tasks, or tasks which the target trainee population has already mastered.

With information about the target trainee population in mind, those job tasks and subtasks which should be included for training are determined using the 6-factor selection model. This is used to assess the Difficulty, Importance and Frequency of each Task, as well as the impact of New Technologies, Safety aspects, and potential Learning Difficulties. The responses, from one or more Subject Matter Experts guide the training analysts as to whether the task should be selected or excluded from training.

The tool must be capable of implementing, at minimum, the Difficulty, Importance & Frequency algorithm (DIF). Ideally, the tool must be capable of implementing the project's custom-designed 6-factor task selection model.

Instructional Setting - This step in the ISD process considers whether job aids (such as checklists, placards, and performance support systems) or On-Job Training (OJT), might be more effective than formal training packages in meeting selected objectives. The tool should prompt the analyst for these considerations.

Develop Learning Objectives - The differences between the skills, knowledge and attitudes possessed by the target trainee population prior to training, and those desired after training (often during their productive service), form the basis of a statement of Learning Objectives. Learning Objectives are arranged in a hierarchy of Terminal and Enabling Learning Objectives. Terminal Learning Objectives can be related to a Task which must be performed in the job. Enabling Learning Objectives can be related to Subtasks.

Learning Objectives are written in terms of the Conditions, Cues and Standards which must be achieved at the completion of the course. Supporting Knowledge, Skills and Attitudes may be referenced.

The tool must allow the Training Analyst to copy the Job/Task Performance Objectives which have been selected for training, and to create from them Learning Objectives, incorporating Conditions, Cues and Standards.

Training Media Analysis - The most important key to an effective training system is to match the nature and characteristics of each Learning Objective with the inherent attributes of an appropriate training medium.

For each Learning Objective, primary consideration goes to the types of cues which need to be presented, the types of responses which need to be made, and how these responses are evaluated.

Visual cues required for helicopter maintenance and operator training includes text, static diagrams, animated diagrams, still images, moving images, colour, and three-dimensional views. Sound cues, tactile cues, kinaesthetic cues, and even olfactory cues may also be needed. For example, innocuous smoke is sometimes introduced into the cockpit of flight simulators to provide the first cue of a serious electrical fault in the cockpit!

Trainee responses can include making a decision, speaking to another crew member, activating a switch, or manipulating a control or lever with psychomotor movements.

Evaluation requirements usually include feedback for the trainee's correct and incorrect responses. Task evaluation can be accomplished inherently by the trainee, automatically by the training device, or with the guidance of an instructor. For mission training, post-training briefings are useful, utilising printouts of the track followed, and decisions made.

Other considerations when selecting a training medium include the potential need to create a safer environment to practice critical tasks (e.g. when responding to an engine fire), the need to reproduce the conditions under which the tasks are to be performed (e.g. restrictive clothing, on board a ship at sea), and the need to create an environment which matches the appropriate stage of learning, from initial familiarity with the aircraft system, through learning the steps in a procedure, to practising the entire task.

Each training medium and device has different inherent capabilities for presenting cues, providing feedback, and evaluating responses. Our manual system uses a matrix to correlate the capability of each training medium to achieve each of the required attributes of the Learning Objective. Examination of this correlation can eliminate mismatched training media, and prioritize the remainder. The most cost-effective media mix which achieves all the requirements is preferred.

The training analysis and design tool must be capable of performing a media analysis to select the most appropriate training media from an instructional point of view. The tool must prompt the training analyst for the set of attributes which might apply to the Learning Objective, and present a resulting ranked list from a generic and comprehensive list of modern training media (such as those shown in Figure 1). The analyst must be free to select any one of the applicable media (not necessarily that with the highest score), since cost-effectiveness and course sequencing may be programme constraints. The tool must allow the user to create comments to supplement the audit trail of why a particular selection was made.

Training Device Specification - The ISD process is iterative. Initially, experience may suggest a suite of generic training media as a starting hypothesis. As a result of performing the key steps in the ISD process, outlined above, the appropriate Learning Objectives will be allocated to each training medium, and the detailed specifications of each training device will begin to develop.

The tool must be capable of being used to consolidate the requirements of a hypothetical training device, and to create a specification for the device. This specification must include a list of the physical and functional requirements for each aircraft system and subsystem to be represented on the training device. The tool should prompt the analyst to establish whether each aircraft system and subsystem requires low, medium or high *physical fidelity*, and low, medium, or high *functional fidelity*. Physical fidelity considers appearance, weight, centre-of-gravity, etc. Functional fidelity considers whether the system needs to work. It must be possible to trace each requirement back to Learning Objectives and Job/Task Performance.

Course Development - The tool must assist the analyst in consolidating and sequencing Learning Objectives into Lessons. Each lesson should build on the previous in a logical progression, with a balanced mix of media and instructional settings (classroom, part-task trainer, simulator, and actual equipment). The tool must assist in assembling the Lessons into Modules, and the Modules into Courses. Each Course is targeted for the needs of a particular target trainee population (e.g. Copilot to Captain Upgrade Course).

Test Items - It is essential to assess the required learning via tests of knowledge and performance. Test items must be properly linked to the Learning Objectives, and must use appropriate media. The tool must be capable of developing and storing test items, and showing the audit trail to the related Learning Objectives.

Cost-Effectiveness Studies - The tool should help perform iterative "what if?" studies in which the scenarios and training media are modified to meet external constraints, such as a limited budget.

The tool should be able to compute the cost effectiveness of any given application of the media analysis algorithm. That is, to assess which of a variety of potential alternative training scenarios (each containing different quantities and mixes of training media) achieves the best training effectiveness and trainee throughput for the lowest cost.

Cost-effectiveness measures require a computation of the total amount and cost of each training medium which has been allocated to the lessons, modules, and courses. For example, the tool should be capable of totalling the run-time hours CBT allocated. If the analyst has defined within the tool the estimated unit cost price of each training medium, the tool should present the estimated project costs for each scenario.

Reports - The tool must be capable of generating a variety of standard and custom reports. Standard reports must include: Job/Task Performance Lists, Tasks Selected for Training, Learning Objectives, Media Allocation, plus reports for Courses, Modules and Lessons.

Security - The tool must implement security controls via user identifications and user passwords. User categories should be implemented which match the requirements of a typical ISD project: e.g. Project Manager, Training Analyst, Course Designer. The tool's security system must restrict the functions permitted for each user category.

Revision & Configuration Control - The tool must be capable of being operated in a multi-user networked environment, with a common shared database, and being used simultaneously by a team of training analysts. The tool must be capable of controlling, and tracking all modifications to the training database. The tool must contain configuration control capabilities so as to mark each record with the date, time, and user who last modified it.

Data Extract & Merge - Facilities within the tool should be provided to allow geographically-separate analysis teams to exchange and merge their separate analyses into one database. The tool should provide facilities to mark selected portions of the database as "locked" or "read only".

Computer Platform & Database - The tool must be capable of running on a Personal Computer, under DOS, and Windows. The tool should use a commonly-available relational database to organise and store the data. The tool must be capable of searching, sequencing, retrieving and cross-referencing data within the database, using Standard Query Language (SQL) or SQL-like statements.

Human Interface - The tool must be easy to use, so that users can focus on the training analysis and design tasks instead of spending time learning new software. The tool should have a user friendly interface, and should require only a minimum amount of training. Functions should utilise the minimum number of input actions (key strokes or mouse actions) to perform data manipulation. The training analysis tool must be easy to customize. The tool must incorporate a comprehensive on-line help facility.

Support & Training - The tool must be adequately supported in terms of documentation and training. The tool must be supplied with easy-to-use operating manuals. Initial training must be available. Technical support must be available for the tool by telephone and fax. The tool should be supplied with a warranty lasting one year or more. New and improved versions of the tool should be available at least once per year, together with add-on modules with enhanced functionality.

Market Share - the selected tool should have been used satisfactorily on several similar projects.

CANDIDATE SYSTEMS

We reviewed the following three training analysis & design tools:

MYSTRO Training Analysis and Design Software, from McAboy Yates Corporation in Garden Grove, CA, USA - Mystro is described [8] as "an automated tool for analysis, design and management of instructional programs". Mystro includes the following software modules (Figure 3): Training Analysis & Design; Survey Module (necessary for selecting tasks for training); Media Selection; Revision Control; Import and Export. Training and support are separate packages.

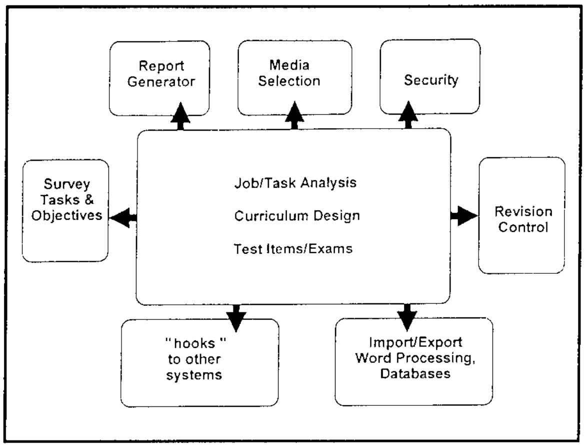


Figure 3 - MYSTRO Functional Overview

Mystro has been used by commercial airlines and airframe manufacturers to manage the Federal Aviation Authority's Advanced Qualification Program (AQP) [11]. Mystro has also been used by a nuclear power company, a telecommunications company, and for a US military application.

JS ISD/LSAR DSS, the Joint Service Instructional Systems Development, Logistic Support Analysis Record, Decision Support System (DSS) - DSS is sponsored and managed by the US Department of Defence (DoD) at Armstrong Laboratories in San Antonio, TX, USA. The tool was developed for the DoD by Dynamics Research Corporation in Andover, MA, USA.

DSS is described [9] as a "major Department of Defence (DoD) effort to better support ISD decision making and to integrate training system development with other weapon system design activities. The PC-based multi-user system consists of data input, ISD analysis, and training system design procedures that reflect and accommodate service-specific ISD methodologies" (Figure 4). A key feature of DSS is the automated LSAR to ISD data interface. DSS includes a variety of algorithms to select tasks for training, and to select appropriate training media.

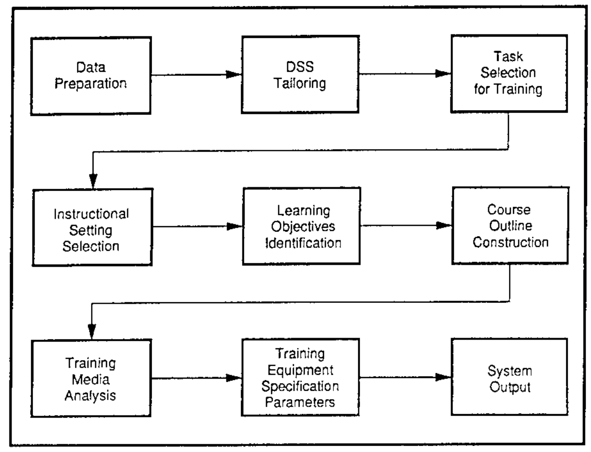


Figure 4 - DSS Methodology Overview

The JS ISD/LSAR DSS has been used by several large aerospace companies to manage the training analysis and design of weapons systems for the US DoD.

The TRACE™ CASE tool for ISD [10] by Trace Technologies Incorporated, in Fayetteville, NY, USA - TRACE was developed as an implementation of MIL-STD-1379D [2] for the FAA's AQP Program [11].

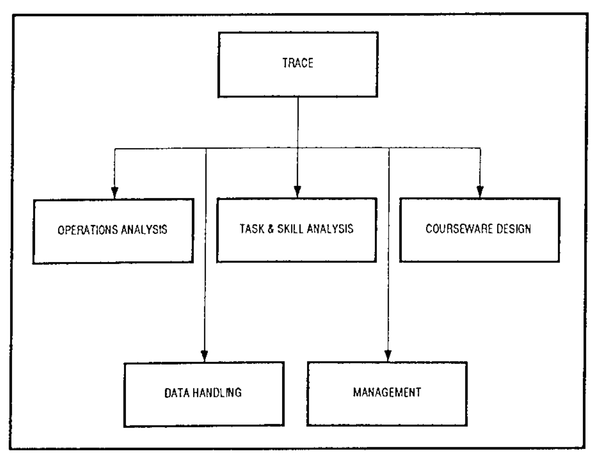


Figure 5 - TRACE Functional Overview

TRACE incorporates comprehensive facilities for Operations Analysis, Task & Skill Analysis, Courseware Design, Data Handling and

Management, including Configuration Management, Database Utilities, Information Management Utilities, System Support and Project Management Tools. TRACE incorporates ISD procedures using the Oracle™ relational database engine and screen utilities.

TRACE has been used in Canada for the training analysis and design for a naval frigate, and is being used by a commercial airline in the USA.

THE EVALUATION PROCESS

The project team used the following mechanisms in its evaluation process for the EH101 Project:

Collate Specifications - We contacted the vendors and asked them to supply technical specifications of their training analysis & design tool, to define version numbers, optional extras, and prices. Vendors were given an opportunity to make presentations on their products.

Compliancy Questionnaire - We devised a questionnaire, based on the requirements outlined above, incorporating 66 statements. Vendors were asked if their package fully met, partially met, or did not meet each requirement. We asked for a description of how each requirement was met. We asked if, and how, any non-compliancies would be overcome by proposed enhancements to the tool, or by additional packages or services.

We asked for estimates of the amount of effort an experienced training organisation would expend on initial user training, and in configuring the package for use on a project with a custom Implementation of the ISD process [5][3].

In-House Evaluation - We arranged a loan of each package, and performed an in-house evaluation, which included a pilot project. The pilot project aimed at testing the most important features of the tool, including:

- Implementing a custom training analysis methodology [5][3].
- Importing a sample of LSA data as a starting point for a maintenance training analysis.
- Constructing a simple hierarchy of Job Performance Tasks.
- Printing a report of the Job Performance Tasks.
- Selecting tasks for training using a simple DIF model.
- Copying the tasks selected for training and converting them into learning objectives.
- Performing a media analysis and allocating media to the learning objectives.
- Printing a report of the Learning Objectives, and media allocated.

References - We contacted references for each tool, and discussed their experiences. We asked their opinion of each tool's strengths and weaknesses.

Spreadsheet - A spreadsheet was used to compile points against each requirement, using the following scale: 0 - Non compliant; 1 - Poorly compliant; 2 - Partially compliant; 3 - Mostly compliant; 4 - Fully compliant.

In addition to points for the requirements outlined, points were also allocated to assess the relative cost of the license, training, and support, together with the in-house setup costs, customisation costs, and operating costs.

A team of evaluators from Westland and the Canadian Prime Contractor allocated the points by mutual agreement. Points were substantiated by reference to source data, such as the compliancy matrix responses, specifications on each tool provided by the vendors, and experience during the in-house evaluations and pilot projects.

The points for each individual requirement were weighted depending upon the relative importance, as judged and agreed by the evaluation team. The total weighted score revealed the best match of requirements for a training analysis and design tool for the EH101 project.

RESULT

All three systems are very capable, and no criticism is intended of any of the candidates. It is not appropriate in this paper to publish the detailed scoring. MYSTRO received the highest weighted score, and was considered the best match between the requirements of a training analysis and design tool for the EH101 project amongst the systems examined.

COMMENTS

One of the main driving factors behind the selection of MYSTRO by the Customer Training School was the need to implement the terminology and requirements of their custom ISD process. MYSTRO does not impose a detailed methodology; system administrators can choose and set up their own terminology, definitions and hierarchies for analysis and design. The methodology and terminology can also be changed from project to project, without programming. MYSTRO scored well for user friendliness.

For projects where a standardised, validated and pre-structured approach is necessary, DSS scores well. DSS is well suited to maintenance training analysis which starts with a MIL-STD-1388-2A or -2B LSAR records. The built-in methodology and terminology are fixed, but the Training Development Manager can tailor the setup for each project, and choose from 6 algorithms to select tasks for training, and 4 algorithms for media selection. Users of DSS do not need to import LSAR records as a starting point, but can input job tasks directly for both aircraft maintenance and operations.

The TRACE tool utilises an industry standard database, Oracle™, which makes it a very good choice where other existing database applications need to be inter-linked with the training analysis and design data.

We considered, but rejected, the option of designing and building a relational database and screens to implement our custom ISD processes. We considered that the costs and risks of such a project would be greater than the use a commercial tool, and considered that it was an advantage to use a well-established and commercially-available tool to set the standard for transferring data amongst organisations and Governments, rather than to develop an isolated non-standard tool, where proprietary considerations might apply.

APPLICATION

At the time of writing this paper (June 1994) the Customer Training School has supplemented its manual ISD process with a computerised process using MYSTRO. The tool is being used in the ongoing analysis and design work for the EH101 Production Investment Phase.

Task and Subtask data from the Logistic Database have been imported and selected records have been used within the maintenance task analysis.

Our custom 6-factor model for the selection of tasks for training has been implemented using MYSTRO, and we have gained considerable productivity over the manual implementation. A custom media selection model has also been implemented with similar savings.

We have tested the exchange of a machine-readable training database containing Task and Learning Objectives. Both sender and recipient used an identically-configured training analysis tool with a common data dictionary. This opened the door to the exchange of data between geographically-remote training analysis and design teams, and provides Westland with the ability to supply the "core" training analysis for the basic aircraft to a Prime Contractor, who could add the additional analysis of the mission systems.

As the project advances, we expect to gain considerable advantage by the introduction of a computerised training analysis and design tool:

- Consistent and semi-automated implementation of the ISD process.
- Detailed guidance for the analysts and designers through the decision making process - less inconsistencies.
- Increased productivity.
- Automatic creation of an auditable record.

- Quicker and easier rework of the training database as part of the ISD process, and as the aircraft system changes.
- The Import of LSAR data without retyping.
- Rapid feedback to update LSA records.
- Quick, easy storage and retrieval of data.
- Shared use of on-line database.
- Reduction in duplicated data and redundant effort.

CONCLUSIONS

The key steps in the ISD process are very labour intensive and rely on a great deal of detailed information from many sources. Commercially available software tools are very effective in supporting training analysis and design by guiding experienced analysts through the required decision making processes: analysts become more productive and make quicker and more consistent training decisions. The tools automatically create auditable and traceable records of the decision processes. Logistic Support Analysis Records (LSARs) can be imported as the basis for a maintenance analysis. Use of (and feedback to) the LSAR helps to integrate the training system development with the evolving aircraft system design. Use of such tools in a networked environment means that training data can be easily and quickly stored, retrieved, shared and exchanged. This sharing of data inevitably results in a reduction in duplicated data, with the associated savings. Provision of configuration control within the software tools allows all modifications to the data to be tracked, and allows the management of changes as the aircraft design evolves. Use of a common software tool, a common data dictionary and database structure allows electronic data interchange of training data amongst organisations in different countries.

ACKNOWLEDGEMENTS

The authors thank all those involved in the project, especially Ms Suzanna Sirignano (Agusta SrL, Italy), Mr Bob Laws & Mr Edgar Guntermann (Paramax Systems Canada), McAboy Yates Corporation, Dynamics Research Corporation, and Trace Technologies Incorporated.

REFERENCES

1. Inter-service Procedures for Instructional Systems Development, Inter-service Training Review Organisation (ITRO), US Department of Defence.

2. Military Training Programmes - MIL-STD-1379D, US Department of Defence.

3. The Systems Approach to Training (SAT). Canadian Government Document A-P9-000.

4. The Systems Approach to Training in the RAF, TSP10, Royal Air Force Training Support Publication.

5. Training Analysis & Design Process - WHL/Agusta Common Baseline. Agusta Sistemi Srl document STD-RD0141, Revision 0, Dated 29 October 1992.

6. Requirements for a Logistic Support Analysis Record, MIL-STD-1388-2B, US Department of Defence.

7. AECMA SPEC 1000D, European Specification for the Identification of Aircraft Systems.

8. MYSTRO Training Analysis & Design Software, Release 2.1, Product Brochure, McAboy Yates Corporation, Garden Grove, CA, USA.

9. Joint Service Instructional Systems Development Logistic Support Analysis Record, Decision Support System (ISD/LSAR DSS), Version 4.3 System Overview, Dynamics Research Corporation, Andover, MA, USA.

10. TRACE, Product Brochure, Trace Technologies Incorporated, Fayetteville, NY, USA.

11. Advanced Qualification Program, US Federal Aviation Authority Advisory Circular AC120-54, 1991.

USE OF "OFF-THE-SHELF" APPLICATION SOFTWARE FOR INSTRUCTIONAL SYSTEMS DEVELOPMENT

Mark C. Stevens
Boeing Defense and Space Group
Seattle, Washington

Gregory S. Davis, Ph.D.
Andersen Consulting
Houston, TX

ABSTRACT

Unlike many training systems that were developed after a weapon system had reached design maturity or even after it was fielded, the F-22 Training System was tasked to be developed concurrent with the weapon system design. Additionally, the F-22 Training System Development Team was challenged to be innovative, look into the future, not accept "non value added effort," to be cost effective and develop an integrated training system. This brought unique analysis requirements. Database and analysis support software was required that could grow with the system, respond to changes in emphasis, data formats and contents, provide insight into the analysis and technical performance, and manage the analysis effort.

A review of existing database and analysis support software built specifically for Instructional Systems Development (ISD) found that none fully met the needs of the program and supported both the pilot and maintenance analysis efforts. It was found, however, that personal computer application software had matured to the point where special purpose software applications could quickly be assembled without special purpose coding, providing a responsive, and cost effective means of managing the analysis effort.

Using the same general ISD analysis methodology, both the pilot and maintenance analysts used "off-the-shelf" software products to acquire, store, manipulate and present analysis data. The major categories of applications included 1) database management 2) decision support, 3) analysis support, and 4) program management tools. We present the results of our efforts to create an integrated local area network environment using commercially available software including software selected, the adaptations we made, and the lessons we have learned to date.

ABOUT THE AUTHORS

Mark C. Stevens is an experienced analyst in Boeing's Defense and Space Group's Military Airplanes Division. He received a BS degree in Aerospace Engineering from the University of Texas at Arlington. While in the Air Force, he flew F-4 and F-5 fighters. He is currently a member of the F-22 Pilot Training Analysis and Integration Team.

Dr. Gregory S. Davis is a manager for Andersen Consulting in Houston, Texas. He has a BS in psychology and a MA in Motor Behavior from Southern Methodist University. He received his Ph.D. in Motor Behavior from The Pennsylvania State University. He is currently working with the ISD analysis team for the F-22 Maintenance Training System.

USE OF "OFF-THE-SHELF" APPLICATION SOFTWARE FOR INSTRUCTIONAL SYSTEMS DEVELOPMENT

INTRODUCTION

Overview

Unlike many training systems that were developed after a weapon system had reached design maturity or even after it was fielded, the F-22 Training System was tasked to be developed concurrent with the weapon system design. This forced us to either build or find ISD analysis software that could grow with the system, respond to changes in emphasis, data formats and contents, provide insight into the analysis and technical performance, and manage the analysis effort.

A review of existing database and analysis support software built specifically for Instructional System Development (ISD) found that none fully met the needs of the program and supported both the pilot and maintenance analysis efforts. We discovered, however, that personal computer application software had matured to the point where special purpose software applications could quickly be assembled without special purpose coding, providing a responsive and cost effective means of managing the analysis effort.

At the beginning of the F-22 Training System ISD analysis effort, it was envisioned that the pilot and maintenance analyses could be conducted in parallel, using the same analytical methodology and analysis software support. Ideally, this would be the most manageable and cost effective approach. As it turned out, the database requirements and analysis approaches differed to the extent that two distinct methodologies and databases evolved. On one hand, where the maintenance analysts received tasks via the Logistics Support Analysis Record (LSAR) and imported the data through interface software, the pilot analysts had to enter all data. There were also differences in the task structure and sequence. Where maintenance tasks were performed more or less independently, the pilot analysts needed to view a task in the context of a series of tasks thus sequencing multiple tasks through the analysis in analytical steps. Maintenance analysts focused primarily on tasks that are performed one at a time; pilot analysts were concerned with the performance of integrated tasks. The two different analysis needs required different database structures and decision support tools.

At program startup, an analysis support package was acquired and tested. After a lengthy evaluation period reviewing several software options, we began to search for ways to develop a set of tools with the "off-the-shelf" software available to us in our work environment.

To meet program milestones, the analyses had to begin. Since the pilot database requirement was small in comparison to the maintenance requirement and was hand entered, Microsoft® Excel™ was chosen as an acceptable alternative. Maintenance software support requirements were more demanding. The software needed to support several users, accept the data in LSA format, and be compatible with the existing Local Area Network (LAN). At the time there were no relational databases that operated in a Microsoft Windows™ environment that were also easy for the end user to learn and use. Fortunately, Microsoft introduced its newest relational database product, Access™, to the market. After attending a workshop on the product, Access was chosen as the software application to support the maintenance analyses.

The "Off-the-Shelf" Environment

One very important benefit to using off-the shelf application software is the ability to continually grow the application in concert with changes in program emphasis and analytical requirements. Seldom does a purchased software package meet all analytical requirements and program management requirements. Either the software must be changed/updated, the methodology bent to fit, and/or the work-arounds constructed. Today's off-the-shelf applications software have matured to the point where customized applications can be easily and quickly assembled to provide analyst and program managers the software support required to meet the dynamics of rapidly changing requirements.

The software applicaton had to be designed to accommodate the iterative nature of concurrent engineering and the instructional system development process, providing for continuous evaluation and revision throughout the analysis, design, development, and implementation. As changes are discovered, either through modifications to the weapon system design or from evaluations of training effectiveness, modifications must ripple down through the training system without significant effort.

The software tools were configured to support a tailored methodology using MIL-STD-1379D and AFI 36-2201 as a guide. The ISD methodology (shown in Figure 1) is divided into segments to support the requirements of the F-22 Training System. We are currently in Segment III, Learning and Media Analysis, and the software was used to support segments I through III. We anticipate that our software needs will develop over time in accordance with our instructional systems development. For example, we see a need for a modeling tool to support economic trade studies and cost comparisons to support the media recommendation process. The advantage to the off-the-shelf tool is the ability to respond to these requirements quickly. The quicker response time allows project management to learn more about the requirements, take advantage of lessons learned, and forecast future requirements more accurately.

Tools Used. The following software applications were used to support a variety of program needs:

Microsoft Word™ 2.0. All of our written documents were produced in Microsoft Word 2.0. We used object embedding and linking (OLE) to attach slides and spreadsheets to the documents. This facilitated reuse of existing documents and consistency in presentation materials.

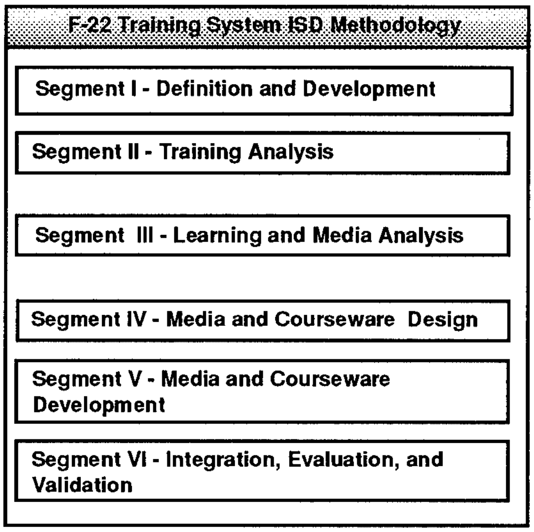


Figure 1. F-22 ISD Process

Microsoft Excel™ 4.0. The pilot ISD analysis and project management metric charts were developed in Excel spreadsheets. The ability to export Access queries and tables to Excel was very helpful to maintenance analysis. The workbook capability facilitated the grouping and linking of spreadsheet data.

Microsoft PowerPoint™. Briefing charts for meetings were created in PowerPoint. We used the OLE capability to update charts built in Excel.

Microsoft Access™. We used Access for the maintenance ISD analysis. It allowed end users to develop a useful, custom application in a few months. We expanded the application's functionality as required. This allowed us to meet our schedules and build the analysis capability as the work progressed.

Unique Considerations

We had several requirements that made the project somewhat unique in that we were trying to make the most of what other programs had used in the way of decision support tools and methodologies. Also, working in a team environment required us to share information quickly with one another using electronic formats. To our knowledge, this is the first program that has combined the engineering of a training system concurrent with the engineering of the weapon system, teaming, technical performance measurement, and electronic information technology all as important components of the program.

Briefing and Status Review Support. There were several occasions where we were tasked to provide accurate updates on the results of our analysis and our progress within hours of receiving the request. Without an integrated set of tools and a mechanism for quickly retrieving data through reports and queries we would have been less responsive to our customer and team member needs.

Decision Support Tools. We embedded several decision support tools in the applications. They allowed us to build in guidelines to improve the overall quality and consistency of the analysis. Further, because we could document the assumptions of the models, it was not necessary to document the rationale for every one of our training decisions.

Application of the Tools to ISD. We tailored the software to the specific needs of pilot and maintenance ISD analysis and this is reflected in the following two sections. We discuss them separately only to highlight the differences in the application of the methodology. We continue to believe that to build an integrated training system, it takes a common methodology. But, when it comes to the specifics of implementing, the methodology tools and models need to be tailored.

Pilot Training ISD Analysis

Initial front end analyses required the development of a pilot task database. Since task data was going to be hand entered and virtually created in real time in the minds of the analyst, it was necessary to have textual software support that allowed scrolling so the tasks could be viewed in sequence and then rearranged and edited as necessary. Also, additional analysis data (conditions, cues, standards, skills, knowledges, etc.) would later need to be defined and entered for each task. It was determined that data entry requirements, content, and analysis requirements could be supported in a spreadsheet environment. A set of spreadsheets were patterned from the pilot task breakdown structure. To manage the development of the database and assess analysis status, summary spreadsheets were developed. Resident analysis software in Excel (Variance, Regressions, etc.) was used to support analyses. Graphing and graphics support in Excel were used to present results.

Pilot Analysis Decision Support Tools

To increase the analysts productivity and to add consistency to results, decision support tools were developed. The built in features within Excel made the development of these tools very easy and user friendly. An example of the Difficulty-Importance-Frequency (DIF) analysis support application is shown in Figure 2. The DIF analysis is used to decide what level of training a task requires. The basic DIF model was modified to include three levels of frequency. Additionally, the analysts wanted to determine early in the analysis the importance of safety, mission objectives, time criticality, and situation awareness for each task. These features were added as Yes/No buttons in the application. The analyst could view the task and related information from the task analysis worksheet in the lower window and use the mouse to select difficulty, importance, and frequency rating in the upper window. If the task had safety, mission, time, and/or situation awareness implications, they too could be selected. The analyst could select "Decision" to see the resulting training recommendation, select new values, and then select "Export" to transfer the ratings and results to the task worksheet. The analysts name and the date were also transferred. As an aid to the analyst, help prompts were included. As shown in Figure 2, the question marks (?) could be "clicked" and a window would appear providing variable definitions. This feature greatly aided in achieving consistent results among multiple analysts. The decision logic, buttons, help

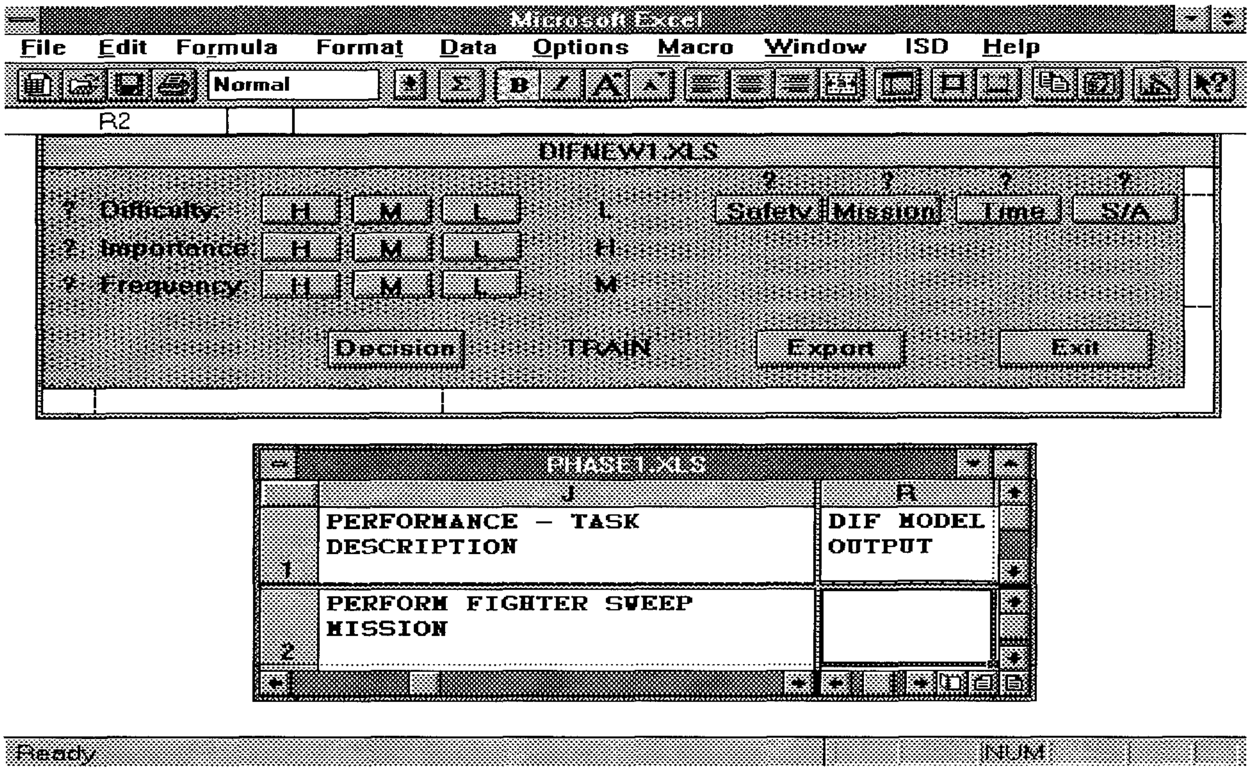


Figure 2. DIF Model in Excel

windows, and data transfer functions were all controlled by macros.

Analysis Support

The pilot database was organized around a typical fighter mission. The mission was divided into seven phases of flight, and each phase into segments. Each segment contains the tasks, subtasks, and activities required to be performed in that segment. Each phase was contained in a single worksheet. There were 9 worksheets. Additionally, there were three decision support tools, status and summary worksheets, macros to generate analysis reports, and worksheets to produce program technical performance measures. To aid the analysts, an "Auto-Open" feature was used to produce a menu specifically for ISD analyses. Dialog boxes were used to access the spreadsheets, summaries and reports. These features in Excel allowed analysts to easily customize the database and make it very user friendly.

Macros were used extensively for counting data items, summarizing, and producing reports. Resident analysis software features were used to perform statistical operations.

Database Management

During database development, configuration of the spreadsheets sometimes became a problem when individual records were either added or deleted. Links to other spreadsheets would sometimes be destroyed. In fact, the dependence on links between data items became a worsening problem as the database grew. Rather than overload the database with links, we were forced to continually add columns to the spreadsheets. The spreadsheets soon became unwieldy. To ease the problem of data entry, the 9 phase spreadsheets were divided into segments. This resulted in 95 spreadsheets. It solved the data entry problem but worsened the link problem.

Maintenance Training ISD Analysis

We faced several challenges when trying to find a workable solution to the problem of conducting a front-end instructional systems development analysis for maintenance training. First, we had to interface with task analysis data from the LSAR. Second, we had to make sure that changes which occurred in the LSAR systems were reflected in our analysis tool. Third, we had to begin our analysis in a few months to meet our scheduled delivery dates for the ISD analysis. Finally, we had to develop a process and a set of solutions that fit a methodology tailored to MIL-STD-1379D and allowed us to efficiently conduct the ISD analysis.

Features of the tool set. The ISD tool has the following features that added user friendliness to the application, increased the productivity of the analysts, and improved the quality of the analysis.

- text file interface to upload and download data from and to the LSAR
- instructional design models for training needs assessment and media selection
- data search keys to find data in the database
- custom reporting
- master pull down lists
- cut and paste capability
- data export to other applications such as Word and Excel

Computer Assisted Manual Analysis (CAMA) Procedures

We named the set of analysis software tools the Computer Assisted Manual Analysis (CAMA) procedures. These tools provided a mechanism for the following:

- encoding instructional design expertise in decision support tools
- storing the results of the analysis
- formatting the results for management reporting and presentations

Most of the analysis work for maintenance training was conducted using several Microsoft Access 1.1 relational databases linked together through table attachments. The host database contains:

- the interface between the LSAR systems both to upload and download data
- tables to store the analysis product

four major functional areas: Task Analysis, Learning Analysis, Media Analysis and Course Design.

User Interface. Access operates in a Microsoft Windows environment and supports graphical user interface features such as buttons, pull down menus, list boxes, combo boxes, and other features which reduce the amount of text data entry. We soon learned that much of the instructional analysis involved the assignment of relatively typical categories to a large number of data elements. For example, there were only a small number of instructional settings that needed to be assigned to thousands of learning objectives. Our solution to this challenge was to use master lists of items extensively to reduce the amount of keyboard entry. By reducing the amount of typing we could conduct the analysis faster, make fewer errors and attain a more consistent product.

Queries. Queries were used to build relationships among tables for both screens and reports. We used queries to understand relationships among data elements and to get a better view of the training system as it developed. For example, we could ask questions such as, "How many learning objectives will be accomplished using a procedures trainer designed for 5-level avionics technicians?" We were also able to produce summary charts such as the percentage of tasks requiring training vs. the percent which required no formal training. We saved several hundred man hours because we were able to ask these questions. This resulted in reduced administrative effort, better quality and less rework.

Functional Overview. The analysis tool contains five major functions: 1) interface with the logistics database, 2) task analysis, 3) learning analysis, 4) media analysis, and 5) course design.

Logistics System Interface. The LSAR data as defined by MIL-STD-1388-2B is stored in an F-22 team Logistic Information System called Systems and Logistics Integration Capability (SLIC2B) (Integrated Support Systems, Inc., 1992). We use a flat file of LSAR tables and import the necessary data into CAMA. Most of the data comes from the CA, CB, CC, and CD task analysis tables.

Task Analysis. Task Analysis begins with an analysis of maintenance tasks to determine the amount and type of training necessary to learn the task. We use the DIF task analysis model to support the training needs assessment (Department of the Army, 1990). The ISD analysts rate the tasks on difficulty, importance, and frequency to arrive at an overtrain/train/no train decision. The analyst can either override the model and document the reason or record the general type of training required: Class, Class and OJT, OJT, or No Training.

The task analysis function also has screens that allow the user to further document the details surrounding correct performance of the maintenance task such as the conditions, initiating cues, and standards.

Learning Analysis. Learning analysis involves:

- the identification of the skill and knowledge required to be able to learn a task
- the development of learning objectives to assess skill, knowledge and task performance

Skill and Knowledge Identification. CAMA allows the user to map skills and knowledge to a task by using a pull down list. This list is created from a master list of skills and knowledge. The master list approach eliminates a great deal of typing and repetitive data entry.

Learning Objectives. We spent a great deal of time considering a variety of designs for developing learning objectives. Our dilemma centered around two issues. First, the relationship of learning objectives to tasks, subtasks, skill and knowledge requires a many to many mapping. For example, one task could require many learning objectives, and, on the other hand, one learning objective could be used to measure the learning of several similar tasks. While we tried to simplify the relationships by clearly mapping the objectives to knowledge, skill, task, and subtask components, there were times where we needed to parse tasks into logical organizational groupings of subtasks. For example, you might use a part-task technique to split the task into setup, maintenance, and follow on activities. We wanted to explicitly capture these arrangements and link them directly to the logistics requirement.

Second, we thought a great deal about the use of terminal and enabling objectives and their relationship to the task hierarchy. We decided to avoid labeling the learning objectives and developed a master list of objectives that could be mapped back to the tasks, subtasks, skills and knowledge. Figure 3 shows the screen that the analyst uses to create a learning objective and map it to a maintenance task.

To build the learning hierarchy, we developed a course structure that was organized into courses, blocks, lessons and modules of instruction. We could then map the learning objectives to the appropriate level in the hierarchy. For example, a terminal objective is achieved when the student is able to accomplish the objective at the end of a lesson. Enabling objectives are satisfied by completion of the module objectives.

The linking approach allows us to take advantage of the relational characteristics of the database and reduces the amount of data entry required. We have also found that our analysis is much more standardized and consistent. Most importantly, it ensures the learning requirements can be traced back from the courses to the aircraft design. So when design changes that impact maintenance occur, the impact on the training design can be assessed. We do this by querying the learning objectives tied to either the logistics data or our instructional analysis.

Media Analysis. Media Analysis includes the selection of instructional media/methods and the recommendation of functional characteristics for training devices.

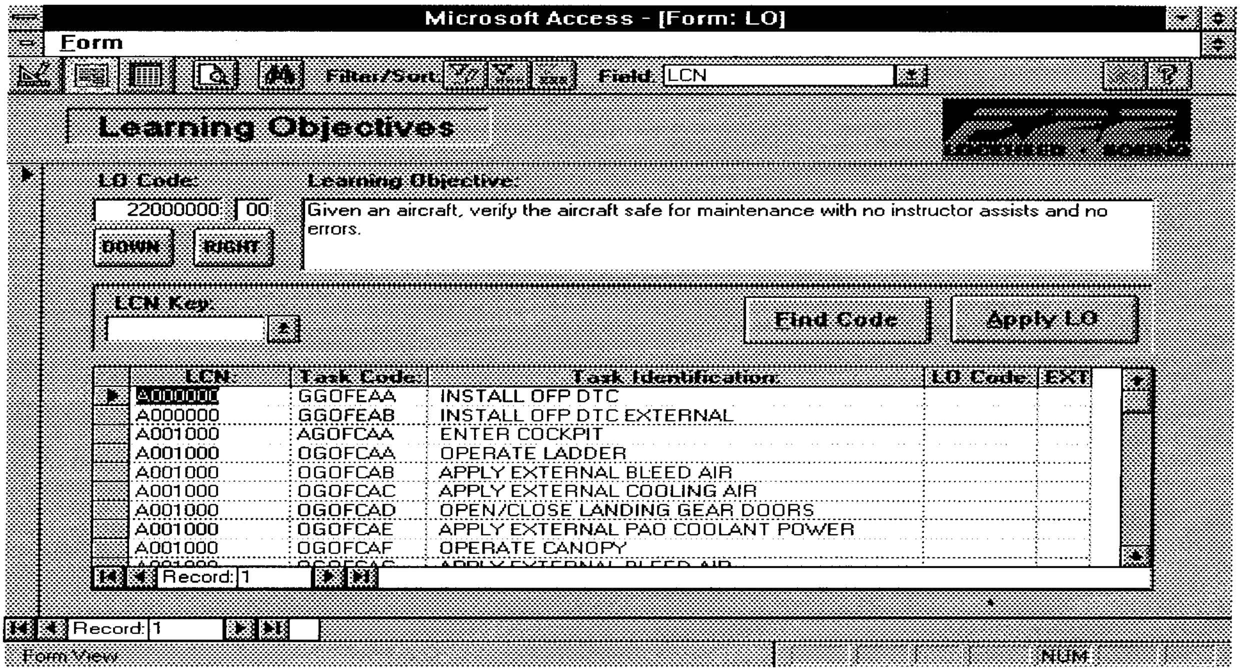


Figure 3. Learning Objectives Entry

Media and Methods Selection. Media selection analysis identifies the most appropriate media to be used for training each task, skill or knowledge by assigning media alternatives to the learning objective. We used a modified version of the Automated Instructional Media System (AIMS) (Kribs, Simpson, and Mark, (1983). We developed a set of tables that contained learning attributes, media and weights which defined the appropriateness of a particular medium for each learning attribute. The analyst indicates which learning attributes an instructional medium should possess to train the knowledge, skills or tasks. The media model produces a list of alternatives, ranking the most appropriate medium first and the least appropriate last. Training methods are then selected from a pull down list. The categories of methods include: information presentation, interaction, and feedback. For example, the analyst might pick a demonstration for the presentation, simulation for the interaction, and outcome feedback as the feedback method.

Training Device Functional Characteristics. CAMA has three major functions for determining the functional characteristics of training device media, such as part task trainers, systems trainers and mockups. These are:

Learning Objective Assignment. Learning objectives can be assigned to trainers after the analyst has determined that a training device should be used.

Fidelity Analysis. This section allows the analyst to list the system components that the student must interact with to demonstrate accomplishment of the learning objective. The analyst also determines the physical and functional fidelity, or realism the component must have to properly learn the task, skill or knowledge. To do this, the analyst rates the fidelity level of each component and then identifies attributes, such as size, shape, center of gravity, etc.

Instructional Features. Each training device can have a set of instructional features. These features include reporting, interaction control, augmented feedback features, data storage and other important features.

Other database functionality included:

Other Databases. We developed several other databases for specific needs not directly related to the analysis but that allowed us to integrate the analysis with project management functions.

Tasktrak. This database is used for tracking completion of the analysis products by aircraft subsystem and analyst.

Bigbro. This database is used for running queries that check the quality of the analysis by looking for noncompliance with project standards, unexpected relationships and inconsistencies in the analysis. For example, if the analyst overrides the training model decision there should be an explanation of the reason. We can run queries that identify these overrides and inspect each case to see if the decision was appropriately documented. This capability is especially important because we expect that we will have over 100,000 important work units in the database by the time we make media recommendations.

Workplan. This database is used to record the work hours expended by the project team. Hours are recorded for each day, person and task. This database also records the results of any revisions made to the CAMA database. The original field, new field, time, date, analyst name, and reason for the change is stored.

Library. We use engineering documents and drawings to conduct the analysis. We built an on-line card catalog to help us store and retrieve information about the documents used.

Archive. One of the challenges of concurrent engineering is dealing with changes in the analysis. The archive database stores data that has been deleted from the logistics system. This allows us to monitor changes and avoid areas that are in flux. If our analysis products get deleted and later a similar logistics requirement gets readded to the system, the analyst can retrieve the prior analysis and reuse it if necessary.

Comments. The comments database uses attached tables from CAMA and allows reviewers to comment on the results of the analysis. The reviewer records their name, date and comment. Open items are tracked and reported.

CONCLUSIONS AND RECOMMENDATIONS

The lessons we learned from three years of experience with automated instructional system analysis tools can be summarized by the following:

Match the off-the-shelf tool to the job. No one software product allowed us to do everything we wanted to do. Also, based on our assessment of tools in the marketplace at the time, we found that it was better to integrate a tool set of several applications than to try to use a tool specifically designed for an ISD analysis.

Do not place too much faith in your decision support models. While it was clearly advantageous to use decision support, an over reliance on the models can be dangerous. The decision support models were very good at eliminating options that were clearly inappropriate, but were not sensitive enough to discriminate among closely related alternatives. Also, these models are designed to operate independently. More work should be done in the future to integrate models. For example, results from training needs models should flow to media selection and these results to fidelity analysis. In the interim, projects should continue to employ qualified instructional designers, as well as, subject matter experts and decision support tools.

Be prepared to adapt to changing needs, levels of analysis and schedules. We found that while the ISD analysis guidance defines a straightforward methodology for conducting a front-end analysis, there were often several different ways to approach the problem. For example, in the maintenance analysis much of the analysis that is typically done to define the job environment is readily available in the logistics system. As much as we tried to avoid duplication of effort, we had to reset our mental models to accommodate the benefits of a more integrated approach. For the pilot analysis, schèduling and reuse constraints required "quick look" analyses and an early assessment of ground-based vs. air training requirements.

While we still face additional challenges, the lessons learned on this program could benefit others involved in front-end instructional analyses. We believe that future efforts in the area of automated instructional analysis should focus on modular analysis tool kits with standard interfaces. These tool kits should work in operating system environments commonly found in the workplace. They should also be developed using commercial software products that are comprehensible to the average end user.

ACKNOWLEDGMENTS

Microsoft and PowerPoint are registered trademarks of Microsoft Corporation. Microsoft Windows, Microsoft Access, and Windows are trademarks of Microsoft Corporation.

REFERENCES

Department of the Army (1990). Systems approach to training analysis. TRACDOC PAM 351-13, Fort Monroe, VA.

Integrated Support Systems, Inc. (1992). Systems and Logistics Integration Capability Users Guide. Clemson, SC: Integrated Support Systems, Inc.

Kribs, H.D., Simpson, A.C. and Mark, L.J. (1983). Automated instructional media selection (AIMS). NAVTRAEQUIPCEN Technical Report 79-C-0104-1. Orlando, FL: Naval Training Equipment Center, October 1983.

THE USAF T-3A TRAINING SYSTEM: NEW DIRECTIONS IN FLIGHT SCREENING

Lt Col James Mohan and Major John Paterson
619th Training Support Squadron
Randolph AFB, TX

Abstract

The 619th Training Support Squadron (AETC) received formal direction to develop the T-3A training system in the spring of 1993. The tasking included the development of a comprehensive training system including aircraft sorties, ground training missions, and academic training in the area of aircraft systems, basic aerodynamics, and flight physiology. The 619th was also directed to provide all supporting materials for these topics such as how-to information on aircraft systems and maneuvers, and audio-visual materials used in classroom presentations. This task was begun even though the air vehicle was not readily available for view and flight manuals were in various stages of development.

At the same time, the AETC Requirements and Acquisition Division requested the 619th provide feedback on a new Air Force Handbook, AFH 36-2235, Volume 8, *Information for Designers of Instructional Systems – Application to Aircrew Training*, the new instructional systems development handbook. Merging these tasks, the T-3A development team relied heavily on the Aircrew Training volume, making a special effort to follow its recommendations.

This paper describes the fielding of the T-3A Training System. It examines the process prescribed in the handbook and how its use affected the development of the training system. The examination will include descriptions of development tools derived from the handbook and the decision making processes. It will also examine the task analysis, media selection factors and decisions, and the results of the analysis. It reviews systemic and personal interactions that both advanced and hindered the development of the T-3A training system. Among those was the limited availability of subject specific information such as aircraft flight manuals and operating limitations. Finally, the paper will describe the finished product including the syllabus of instruction and courseware. It will also include feedback from the students and instructors engaged in this new program.

ABOUT THE AUTHORS

Lt Col James C. Mohan is the Commander of the Training Technologies Flight of the 619th Training Support Squadron (AETC). He has a Masters degree in Management from Golden Gate University and a Doctorate in Vocational and Adult Education from Auburn University. Lt Col Mohan has over 13 years experience in USAF pilot training principally in primary jet training. He has served as an Undergraduate Pilot Training Instructor Pilot, Flight Commander and Operations Officer. He was also the curriculum director for the Air Force Junior ROTC program.

Major John Paterson is the Program Manager for the T-3A training system at the 619th Training Support Squadron (AETC). He has a B.S. in Engineering Sciences from the US Air Force Academy and an M.S. in Aeronautical Science from Embry-Riddle University. He has experience as an EC-135 instructor pilot and over six years in the Undergraduate Pilot Training arena as a T-38 instructor pilot. He is currently qualified in the T-3A.

THE USAF T-3A TRAINING SYSTEM: NEW DIRECTIONS IN FLIGHT SCREENING

Lt Col James Mohan and Major John Paterson
619th Training Support Squadron
Randolph AFB, TX

INTRODUCTION

Over the years the United States Air Force has used many ways to screen pilot candidates. The purpose of this screening is to identify those unlikely to succeed before thousands of dollars are invested in their training. This has become even more important as training costs have grown. Primary flight training in the T-37 aircraft costs nearly $400 per hour and more than twice that amount in the high performance T-38.

Flight screening has taken many forms in the recent past. Air Force Academy cadets currently receive screening through the Academy's airmanship program using the Cessna T-41 aircraft. Officer Training School pilot candidates have screened at the 1st Flight Screening Squadron (1 FSS), now 3rd Flying Training Squadron (3 FTS), at Hondo Airport TX. AFROTC cadets have been screened through programs at local fix base operators (FBOs) but are now also screened at Hondo, usually between their junior and senior year of college.

There have been numerous studies of the screening process. The Fairchild Library at Maxwell AFB AL has numerous studies and reports describing possible screening plans and the cognitive and psychomotor tasks screening should evaluate. Most recently, the Air Force Armstrong Labs have completed exhaustive studies on successful and unsuccessful candidates. In the lab's research, candidates took computer-based exercises on both specially constructed and Z-248 computers. Additionally, students underwent interviews conducted by trained interviewers.

The goal of this most recent research was to support the Air Force' movement to Specialized Undergraduate Pilot Training. It was hoped that together with flight screening, candidates could be screened not only for flying aptitude, but also further screened as to suitability for the fighter/bomber track or the airlift/tanker track. Air Force leadership decided to delay this tracking decision until after primary flying training, however.

The leadership also decided that basic flight screening in the T-41 aircraft was insufficient to meet the increasing complexity of UPT and SUPT. They wanted the flight screening aircraft to have the capability to perform basic aerobatics, spins, and 360 degree overhead traffic patterns, unlike the box pattern of the T-41.

Aircraft Acquisition

Air Force specifications for the enhanced flight screener (EFS) aircraft called for a commercially available Federal Aviation Administration-certified aerobatic aircraft. It needed side-by-side seating and dual, stick-type controls. Furthermore, it needed a piston-driven engine capable of delivering normal cruise speeds of 155 knots. The system acquisition did not call for the purchase of the supporting training system components such as syllabus and courseware.

In April 1992, the Air Force selected the Slingsby *Firefly* as the new flight screening aircraft. The British designed aircraft would be assembled by Northrop Worldwide Aircraft Services at Hondo, Texas. The aircraft would be missionized for Air Force needs. An example of this missionization is the larger engine installed in the Air Force version of the *Firefly*. The engine is supplied by Textron-Lycoming and the avionics suite by Bendix-King. The Air Force has designated the aircraft as the T-3A.

TASKING

The 619th Training Support Squadron (AETC) received formal direction to develop the T-3A training system in the spring of 1993. The tasking included the development of academic training in the areas of aircraft systems, basic aerodynamics, and flight physiology. The 619th was also directed to provide supporting materials for these topics and how-to information on aircraft maneuvers and operations. Responsibility for this last area of instruction was to be shared with command standardization and evaluation officials. The Stan/Eval division has traditionally maintained management responsibility for how-to-fly books used by AETC students.

At the same time this tasking arrived, so did another one on an entirely different subject. This tasking dovetailed neatly into the T-3A project and became the backbone of the training system development process. The AETC Requirements and Acquisition Division requested the 619th to provide feedback on a new Air Force Handbook on instructional systems; in this case, AFH 36-2235, Volume 8, Aircrew Training. This handbook was part of a comprehensive project to update Air Force instructional systems design (ISD). This project's overall goal was to simplify ISD guidance. The method chosen was to develop handbooks dedicated to specific interest areas. For example, volumes exist for Aircrew Training, Technical Training, Interactive Courseware as well as other training-specific areas. The development team relied on the Aircrew Training volume heavily making a special effort to follow the recommendations outlined in the handbook. The ISD model from the handbook is shown in figure 1.

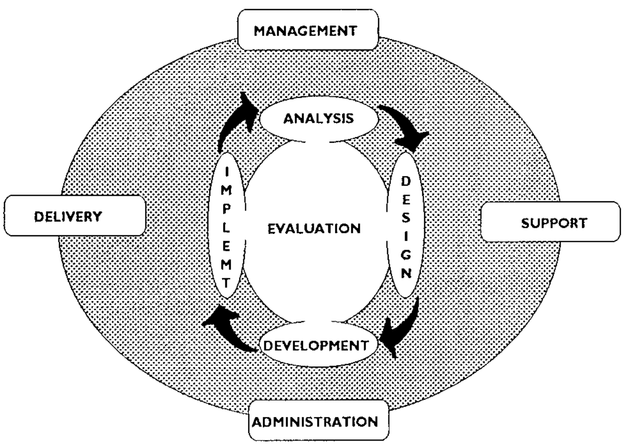


Figure 1
The Air Force ISD Model

PROJECT DEVELOPMENT

Preplanning ISD education

Course maintenance tasks make up the vast majority of training management tasks faced by the 619th TRSS. As such, ground up ISD task accomplishment is seldom required. While all designers in the squadron are familiar with the process, few have gone through it from start to finish. This, together with the new Air Force handbook team members were testing, resulted in a considerable amount of process review and, in some cases, convincing participants that early ISD steps were indeed important. Each participant reviewed the Aircrew Training volume before any initial meetings took place. The team dedicated the initial meetings to developing a mutual understanding of the process on which they were about to embark. All questions concerning the need to do background research and planning were addressed and settled. While some skepticism existed, all members quickly realized that the development of a comprehensive program would turn to chaos without the structure called for in the handbook.

Team members were also trained in the development of criterion referenced objective writing and techniques for arranging objectives in task hierarchies.

Planning

Training concept. In the meetings dedicated to planning, the development team addressed the following questions:

- Why are we developing this training?
- What type of instruction do we anticipate?
- What do we want our training needs assessment to accomplish?
- How much instruction do we anticipate?
- What tools do we need to complete the tasks?

The answers to these questions helped focus on the major factors essential to the development process.

Why train? The answers to the question 'why are we doing this' were straight forward. First, the Air Force has a continuing need to screen pilots before Undergraduate- and Specialized Undergraduate Pilot Training (UPT). Second, using a new aircraft with significantly increased capabilities, eliminated simply repackaging the current training program as an acceptable option. Additionally, since the current program evolved over many years, no instructional design history was available. This provided us with the opportunity to examine the system from the ground up. Lastly, as noted before, the acquisition strategy did not include any training system support.

Needs assessment goals. The development team decided to clearly identify all externally directed tasks (headquarters requirements), user identified tasks, and shortcomings of the previous program. Also, time plays a very important role in flight screening. Trainees have only 25 training days to accomplish the training. Extensions are difficult to acquire. Additionally, air staff planners program flying hour funding based on strict guidelines. The needs assessment had to clearly identify these limits.

Anticipated instruction. The team also looked at possible types of instructional delivery. Since no moneys had been set aside for equipment purchases and because the student flow at flight screening is very cyclical, computer based training (CBT) was rejected. The possible instructional options identified were: in-the-aircraft, classroom, instructor pilot one-on-one briefings called P-missions (P = procedures), student workbooks, and cockpit procedures trainers (CPTs).

Tools. Since the 619th had not developed a complete training system recently, the development team examined a number of tools to assist us in this task. Team members determined that the project was a fairly simple one. They anticipated 300 to 500 objectives for the T-3 program. It was also known that CBT and simulators would not be part of the training media mix. With that in mind, several automated design tools and media selection models were rejected. Team members did realize the need for a computer data base for objective management. The need for paper worksheets and data collection tools addressing points in the aircrew training handbook was also identified. Those familiar with the new ISD model realize that planning is not one of the items listed in the graphic of the model. It is however, a key element of the quality improvement foundation on which the ISD model rests.

Analysis

Process. The analysis phase of the project focused on several questions. In addition to the issues included in our needs assessment concerns in the previous paragraphs, developers wanted a specific listing of required training tasks and an analysis of our target trainee population.

Basic needs. The training needs assessment identified both the headquarters and user identified training requirements. Most noteworthy, was an Air Force Chief of Staff directive that trainees should accomplish a solo flight to the air work area in addition to a pattern-only solo. While some staff agencies expressed concern over this addition, it became clear that trainee confidence is an import aspect of what the Air Force is trying to screen. The development team felt a confidence-challenging sortie such as the one directed would certainly provide some insight into this important aspect of future trainee success. Developers also judged the addition of basic aerobatics and spin training to be valuable additions to the screening program. These maneuvers help assess trainee adaptability to stress, trainee confidence, and physiological adaptability to multiaxial accelerations and G forces. Due to its low wing configuration, the T-3A aircraft also allows trainees to fly 360 degree overhead, jet-type traffic patterns. This complex task with early-required proficiency, provides an excellent opportunity for screening.

Instructional options. The needs assessment also confirmed the available instruction options. CBT was ruled out. Cyclical student load did not justify the use of specialized computer assets. Flight screening is heavily weighted to the summer months since most of the trainees are college students who are in class during the rest of the year. Class size in the winter is sometimes as low as 6 or 7 and as high as 25 to 30 in July and August.

Initiatives were made to fund cockpit procedures trainers (CPTs) and to improve the multimedia capability of available classrooms. After analyzing training tasks and objectives (discussed later) the need for a CPT became clear. Working with the AETC trainer fabrication shop at Randolph AFB, the team developed funding requirements and acquisition strategy. It also discovered shortcomings in the presentation capabilities of the flight screening classrooms. Working closely with the 12th Flying Training Wing, parent wing of the 3rd Flying Training Squadron, a multimedia upgrade to the main classroom was proposed. Team members were certain that training support materials would include such things as video tape and still images, as well as text slides and drawings. A presentation system that allowed integration of video in a window, video stills capture, quick and easy development of text slides, and the ability to project digital still images was selected. Working with the 12 FTW resource advisors and command visual information specialists, plans were made to purchase multimedia production and presentation equipment for the T-3A program. The equipment was chosen based on plans to test the same equipment in both technical training and pilot training. Money was budgeted for this purchase and the equipment was obtained.

External limitations. Flying time and calendar time limits were also confirmed. Flying hours were restricted to 21.5 hours per student and training days were limited to 24. Five in-processing and administrative days precede training.

Target population. Target population analysis was accomplished by interviewing current and former flight screening personnel. The analysis indicated that the trainees using the training system would be a tremendously diverse lot. They would not only vary in age but also in previous flying experience. As stated earlier, the traditional trainee is a college junior or recent graduate. He or she will most likely be ROTC or academy cadet. Some will be recent graduates of Officer Training School. Total numbers from each of these sources are difficult to determine due to the current force drawdown plans and their impact on pilot production. In addition to these trainees, there is a possibility of older trainees who have been selected from other Air Force career fields; although this number is small.

Trainee flying experience varies tremendously. Many of the trainees have FAA private pilot licenses. In fact, many trainees believe that obtaining this rating prior to flight screening will help them successfully complete the program. In some cases, flying experience is extensive, especially among those candidates coming from Officer Training School. It is not surprising to learn that an OTS candidate has experience as a commuter airline pilot or as a corporate pilot with an FAA commercial instrument or even an airline transport pilot rating. This experience helped trainees complete the course in the past. It is anticipated that flying experience will be less of a factor in the T-3A. Air Force officials believe, however, that the introduction to a rigid, regimented training program benefits all UPT and SUPT candidates. While basic airmanship screens may be less effective, adjustment to military life and self discipline screens are still effective.

Training tasks. During the training task analysis the existing flight screening syllabus, course training standards, available T-3A flight manuals and common Air Force and AETC flying regulations were examined. The team developed a list of activities trainees needed to accomplish and grouped them into nine distinct areas of flight training. These were:

- Ground operations
- Takeoff and departure
- Basic aircraft control
- Advanced handling -- stalls
- Advanced handling -- aerobatics
- Descent and recovery
- Pattern operations
- Flight management
- Emergency procedures

Each of the major events that occurred in the above mentioned tasks was then identified. These events became the top level training objectives making up the T-3A training system. At the end of the initial review, 66 such tasks were noted. As the training hierarchy developed, over 400 mid- and low-level objectives emerged.

Results. At the conclusion of the analysis phase a firm training system foundation was in place. Training task identification and media selection had been completed and an outline for the final course was emerging. Most significant issues had been identified and actions needed to deal with them begun.

Tools. To assist in this phase, job aids and worksheets were developed. These worksheets helped focus attention on the key issues. These issues included training needs identification, task identification, task analysis, target population analysis, and resource requirements. These worksheets were direct descendants of the job aids provided in the ISD manual. The manual provided a clear outline of the issues that needed to be addressed. While some were obvious, examining several of the less obvious ones resulted in important contributions to the total training program. In retrospect, inclusion of those less obvious issues in the analysis resulted in discussions that significantly altered the final appearance of the training program.

Design

Process. The design phase of the project consisted of three supporting tasks and the establishment of the design itself. The supporting tasks were the development of objectives and tests, a review of existing materials and procedures, and the examination and selection of alternative training methods.

Objective development. Training objectives flow from the training tasks. Each task was examined and divided into its component parts. These parts were systematically divided into smaller and smaller parts until bite-sized events remained. For example, in the area of take-off and departure, a top level objective is to navigate the aircraft tô the air work area. The standard is to use local procedures and ground references. The next objective down the hierarchy is to point out selected ground references on the ground. Moving further down the list the objective demands trainees match symbols on a map with descriptions or photos. In this manner, objectives were formed to support the nine top level tasks. Final objective count totaled over 400.

A PC database was used to store, retrieve and arrange training objectives. The database included information on each objective including task, date, objective number, media, condition, capability or behavior, and the standard. The database program allowed the retrieval and grouping of objectives by any of the above mentioned categories. As the older database software program became cumbersome, the entire database was transferred to Microsoft *Excel*. The easy sorting and printing options far surpassed the older database program.

Each objective was evaluated as to its appropriateness and conciseness. In the initial review, the subject matter experts recommended deleting a number of the objectives. These recommendations were based on the continuing concern of over *training* in what is supposed to be a *screening* program. Determining what was required and what was nice to have became an important aspect of the design phase.

After an initial examination of the objectives, each objective's training media was examined. As discussed earlier, media choices were limited to the aircraft, classroom presentations, texts and workbooks, and the possibility of a cockpit procedures trainer. During this initial media selection process a CPT was assumed. Objectives were then sorted as to proposed media. The categories were:

- Aircraft
- Flightline (Instructor pilot briefings and workbooks)
- Cockpit procedures trainer
- Classroom (texts, video tape, slides, lecture)

Figure 2 is an excerpt from the training objective database.

Task	Date	O b j c t n	Media	Condition	Capability	Standard
Preflight	3/22/93	1 1 1	Classroom	Given a typical forecast for the local area, flying area, and aerodrome	identify factors which would negatively affect the completion of a planned syllabus mission	without error
Preflight	2/16/93	1 1 2	Flightline	Given flight planning requirements,	obtain operations data for mission planning	IAW FAA, AF, and ATC directives.
Preflight	6/29/93	1 1 2 1	Flightline	Given an open operations center/flight room data display	obtain weather data, and flying status	without error.
Preflight	6/29/93	1 1 3 1	Classroom	Given the terms: runway length, runway slope, headwind component, crosswind component, and tailwind component	Select the definition describing the term	without error.
Preflight	2/16/93	1 1 4	Flightline	Given computed TOLD,	complete a takeoff and landing data summary sheet	without error and IAW checklist and aircraft dash 1 procedures.
Preflight	2/16/93	1 1 5	Aircraft	Given access to the clearance authority,	obtain clearance for a local VFR flight	IAW ATC and FAA directives.
Preflight	2/16/93	1 1 6	Flightline	Given FLIP planning documents and aeronautical charts,	plan a VFR mission	IAW FLIP planning, FAA, AF, and ATC directives.
Preflight	3/26/93	1 1 6 1 1	Classroom	Given an aeronautical chart,	point out the following:a. Latitude/longitude lines, b. Magnetic variation, c. Terrain, d. Rivers and bodies of water, e. Various other map symbols	without error.

Figure 2
Excerpt from T-3A Training Objective Database

Test Development. Testing objective accomplishment played a major role in the development of the objectives themselves. As each objective took shape, it was evaluated as to how it would be tested. Specific objectives minimized the effort required to develop tests. For example, when the objective calls for the trainee to label prominent ground references on a blank diagram of the training area, almost no thought is required to come up with the evaluation. The biggest decision in this case was to determine the scale of the map. In developing the academic test instruments, it was decided to sample at least one objective in each lesson segment. Since lesson segments were narrowly defined, such sampling produced a high percentage of the total number of objectives being sampled.

To ensure aircraft and flightline objectives were all accounted for, each objective was matched to the in-flight maneuver or task to which it applied. Figure 3 shows an example of this cross referencing.

Man No.	Maneuver	C16/4	Objectives
01	GROUND OPERATIONS	G+	1.1.1, 1.1.1.1, 1.1.2, 1.1.2.1, 1.1.3.1, 1.1.4, 1.1.5, 1.1.6, 1.1.6.1.1, 1.11, 1.2, 1.2.10.1, 1.2.10.1, 1.2.10.2, 1.2.2, 1.2.2.1, 1.2.2.1.1, 1.2.2.1.2, 1.2.2.1.3, 1.2.3, 1.2.4, 1.2.5, 1.2.6, 1.2.6.1, 1.2.6.1.1, 1.2.6.2, 1.2.6.3, 1.2.7, 1.2.7.1, 1.2.8, 1.2.8.1, 1.2.9, 1.3, 1.4, 1.5, 1.5.1, 1.5.1.1, 1.5.3, 1.6, 1.6.1, 1.7, 1.9, 1.9.1, 7.13, 7.13.1, 7.14, 7.14.1, 7.15, 7.15.1, 7.16,
02	TAKEOFF	G+	2.1, 2.1.1, 2.1.1.1, 2.1.1.2, 2.1.3, 2.1.3.1, 2.1.3.1.1, 2.1.3.1.2, 2.1.5, 2.1.5.1, 2.1.5.1.1, 2.1.5.1.2, 2.1.6, 2.1.6.1, 2.1.6.1.1, 2.1.6.1.1.1, 2.1.6.1.1.2, 2.1.6.2, 2.1.6.2.1, 2.1.6.2.1.1, 2.1.6.2.1.2, 2.1.6.2.1.3, 2.1.6.2.1.4, 2.2,
03	DEPARTURE	G+	2.2.1, 2.3, 2.3.1, 2.3.1.1, 2.3.1.2, 2.3.1.3, 2.4, 2.4.1, 2.4.1.1,
04	CLIMB	G+	2.2.2, 2.2.2.1, 2.2.2.2, 2.5, 2.5.1, 2.5.2,
05	LEVEL OFF	G+	2.2.3, 2.2.3.1, 2.2.3.3, 2.2.3.3.1, 2.2.3.3.2, 2.2.3.3.3, 6.2.4, 6.2.4.1, 6.2.4.2, 6.2.4.3,

Figure 3
Excerpt from T-3A Maneuver Item File

Existing material review. The existing material review uncovered a wealth of usable courseware and supporting information. Material from the existing T-41 flight screening program, Canadian Forces courseware for their version of the *Firefly*, courseware from the USAF enhanced flight screening test program, FAA private pilot study guides and aids and commercial video courses were all examined. Developmental flight manuals for the T-3A provided by the contractor were also available.

The primary purpose for this review was to avoid reinventing the wheel. No one was opposed to cutting and pasting relevant portions of existing material assuming proper copyright releases and permissions could be obtained. Of particular value were the video tapes from commercial vendors. While the information was typical of other sources, the order the material was presented was of particular interest. Presentation mix was a major design concern.

Training alternatives. Resource limitations and constraints identified early in the planning and analysis process resulted in little flexibility in the identification of alternative training possibilities. The only area in which reasonable alternatives existed was in the area of checklist usage and emergency procedures.

Lessons tentatively assigned to a CPT needed to have an alternative in the event of large class sizes (surges) or CPT acquisition delays. The aircraft was identified as the alternative medium. The aircraft was not the primary medium due to the high cost of repairing accidental damage and operational constraints such as keeping power on the aircraft without any intention of starting it (battery life, component wear and tear etc). Additionally, out-year programming indicated high aircraft utilization rates that would not allow free use of spare aircraft.

The 3rd FTS identified a shortcoming in the area of classroom instruction that required alternative planning. Classroom presentation techniques were limited to 16mm film and 35mm slides. Since no 16mm film production was planned, video tape presentations to large groups were required. Plans were established to borrow projectors or to divide trainees into groups small enough to use available TVs. The 3rd FTS with support from the 619th also initiated the proposal to upgrade to basic multimedia in the classroom.

Academic/aircraft mix. Placing academic foundation materials in their proper place in the training flow was a primary concern throughout the training design process. Since much of what was being taught was totally different from what many of the trainees normally dealt with, moving material one day could make a difference in training effectiveness. With that in mind, training objectives whose preferred medium was the classroom, were arranged into groupings that would closely match aircraft sequence of events rather than the more traditional subject matter arrangement found in UPT. Using numerous prerequisites was also rejected.

Instruction in aerospace physiology was placed first since, by regulation, that subject matter had to be covered before students could fly. The limited objectives of the physiology course focused heavily on quality-of-life issues such as sound eating and sleeping habits and their effect on pilot proficiency. Instruction on anti-G straining was introduced during classroom physiology instruction and is reinforced by flight line instructors prior to every flight on which moderate or high G maneuvering is planned.

Academic instruction was divided into lessons on aircraft systems, basic aerodynamics, and flying fundamentals not included in the previous subjects. Items such as map reading and basic instruction on navigation procedures were included in the fundamentals category. As stated earlier, it was decided to mix instruction in each of these areas. Instruction given before the first sortie emphasized those tasks that occur on the first sortie. Besides physiology, preflight instruction included instruction on major system components, the effects of flight controls, thrust, weight, lift and drag, and map reading. Follow-on instruction completed trainees' knowledge of aircraft systems and deepened their understanding of aerodynamics. All academic instruction is complete by the fifth training day. This corresponds to the inflight instruction block in which strong emphasis on stalls occurs. It is also just prior to the presolo block of instruction.

Flightline ground training. A great deal of the instruction trainees receive comes from their assigned instructor pilot in the form of preflight briefings and ground training missions call P-missions (Procedural missions). Ensuring that each trainee received the same basic instruction from his or her instructor was a fundamental concern. The use of special syllabus instructions and mandatory briefing items given as part of a particular mission's preflight briefing was examined and discarded. Instructors stated that there was little time for extended briefings during the course of a two- or three-period flying day. Instructor recommendations to make use of well supported, structured P-missions were accepted as the best way to convey the material. It was decided to support these briefings through student ground training workbooks and comprehensive instructor briefing guides.

Results. As the formal portion of the design effort closed, developers had a well developed outline of the training program and were well prepared to assemble the course control documents called for in the development phase of the model

Tools. Charts and tables in the Aircrew Training manual were used frequently during the design phase of the process. Thumbnail sketches of key elements of learning theories were sufficient to remind designers of the issues they must address. Tables on media selection criteria were particularly valuable since using an automated assessment tool had been rejected. The concise listings of advantages and limitations in one easy-to-use table also enhanced rapid decision making during this phase. The sections on objective development and testing formed the basis of the pre-project training group members received.

Development

Process. The first task undertaken in the development phase was the formal development and approval of the course syllabus. Draft syllabuses had existed up to this point to provide key leaders with an idea of the form the final course would take. The final version had several changes from the preceding drafts. Issues surrounding the area solo flight and pre-flight training in checklist usage were clarified and instituted. Additionally, final maneuver item files (used for listing grading standards and competencies) (Figure 2) were agreed upon. Course training standards were placed in the traditional format and compared to the objective lists to assure consistency.

Courseware. Academic classroom support materials and student workbooks and texts came next. After evaluating other materials it was decided that since Air Force rules required certain material be included in specific publications, the T-3 program would not try to consolidate information from different sources. The primary rational for this decision dealt with courseware currency. When data from sources is duplicated, any change in the original source forces changes throughout the courseware. It was decided that students would be given the tech orders, regulations and workbooks rather than a comprehensive text covering the important aspects of these multiple sources. Workbooks would cover only that information not covered in any other source or information describing the preferred technique when multiple sources offered acceptable but differing guidance. To ease the inefficiencies multiple sources induce, student workbooks listed specific references in the various source materials for study.

Validation. Operational tryouts were conducted in March and April of 1994. Academic courseware included prototypes of the multimedia courseware and final draft editions of instructor guides, student workbooks and flightline briefing guides. Academic instructors practiced teaching the material under the observation of development team members. Obvious disconnects were identified and addressed on the spot. Additionally, development team members observed presentations to the first two classes of students (approximately 10 students total). Students were interviewed after the classes and asked about the session. They were also asked to critique the end-of-course examination. Student and instructor inputs resulted in changes in the time allotted for some subject areas. Students agreed that the end-of course exam was too easy. However, since the exam accurately reflected the desired level of learning and was true to the course objectives, it was not significantly modified. Students and instructors were reminded that in a mastery learning scenario, high scores signify goal attainment.

During the operational tryouts, the multimedia presentations were taking their final forms. By design, their development followed the operational tryouts. Academic instructors from the 3rd FTS provided valuable feedback as to the effectiveness of the original courseware. That feedback became an important criterion in the development of the final version.

One interesting point of conflict in the development of the final version had to do with the use of the technology. The selected delivery system had the ability to present word slides and pictures. However, it also had the ability to present video tape segments from various sources and sound. Since experience in using this type of presentation media was low, first attempts were often slide shows with fancy transitions between slides. Breaking the 35mm slide model of academic presentations was a constant challenge. The idea that a 10 to 50 second video segment was just as effective as a 12 minute segment was difficult to grasp. Developers had to constantly remind themselves to think creatively and not to rely on experiences that were often based on slides and overhead transparencies. This was particularly true of the academic instructors whose experience was limited to linear presentations with almost no technology support.

Results. At the conclusion of the development phase, a fully functioning training system had emerged. The training support materials developed included the flight screening syllabus, student workbook, academic instructors' guide, backup 35mm slide academic classroom presentation materials, primary electronic classroom presentation materials, AETC Manual 3-3, Primary Flying, ATEC Regulation 55-4, T-3 Aircrew Operational Procedures, the aircraft technical order (Dash One), and syllabuses for Pilot Instructor Training and International Student Training.

Tools. The 619th is a fully functional training material publishing house. Word processing, computer graphics, photo retouching and page layout tools were used along with computer generated 35mm slide and electronic presentation equipment. Survey instruments and questionnaires were based on questions included in the Aircrew Training manual.

Implementation

Process. T-3A Training System implementation was designed as a smooth flowing follow-on to the operational tryouts that culminated the development phase. Instructors from the 3rd FTS were instructed on the operation of the training system during the operational tryouts. They were also very familiar with the overall training concept that did not differ significantly from the older T-41 Flight Screening Program. Ground training events were rehearsed with the multimedia presentation system and flightline instructors were instructed on the use of instructor briefing guides used in various phase briefings.

Review. Management controls covering the full array of student training issues were examined. Existing management programs from the T-41 program were deemed appropriate and reinstituted with the arrival of T-3A trainees. The existence of these controls and formal course critique and evaluation programs reduced the total effort required for the program's implementation. As the T-3A Training System became fully operational, the only open items were training device issues. Models of the T-3A were in development and the CPTs were awaiting funding. The existence of training alternatives decided upon in the design phase resulted in no significant degradation to the total training system due to these open issues.

Evaluation

Process. As the ISD model shows, evaluation occurred continually throughout the training system's development. As each task within the phase was completed, members of the development team examined it to determine whether it met expectations and stayed true to the systems' design. Users were involved in the evaluation as well. Flightline and academic instructors examined each training product and their comments were assessed and in most cases incorporated into the materials. Interim evaluations pointed to shortcomings that were addressed before training began.

END PRODUCTS

In its final form, the T-3A training system consists of the following activities:

- 14 hours of classroom instruction on physiology, aerodynamics, aircraft systems and flying fundamentals
- 9.5 hours of flightline ground training (P-missions) covering flightline procedures, trainee responsibilities, flying safety, stall and spin instruction, aerobatic instruction and emergency procedures.
- 1 checklist procedures lesson (CPT or aircraft)
- 1 orientation aircraft sortie
- 6 pre-solo basic maneuver sorties
- 4 pre-solo intermediate maneuver sorties
- 1 supervised solo sortie (pattern only)
- 4 post-solo intermediate maneuver sorties (aerobatics and 1 area solo)
- 1 pre checkride review sortie
- 1 checkride

Total flying time:	21.5
Total ground training time:	25.5
Total training days:	1 preflight & 24 flying

CONCLUSION

The T-3A training system development project allowed the Air Force the opportunity to reexamine flight screening from the ground up. Using the development outline contained in AFH 2235 volume 8, Aircrew Training, ensured all relevant facts were examined and previously held truths challenged. The resulting training system reflects all the training and screening tasks identified through the analysis phase of the ISD process. Throughout this fast-paced development project, the Aircrew Training volume provided invaluable guidance and insight into the process. Team members agreed that the structure of the newly designed ISD manuals allows easy access to the information without having to wade through pages of lengthy explanations. Not only was the choice of information excellent, its presentation in the Information Mapping format ensured rapid access to required facts.

By all early indications, user satisfaction with the T-3A training system is high. Top level managers have access to easily traced training objectives and tasks. This allows them to relate student performance trends directly to identifiable flying maneuvers, ground training units or academic lessons. Instructor pilots and student managers can quickly and easily relate performance standards to individual maneuvers. Detailed instructor guides for both academic and flightline instructors ensure all students receive the same information resulting in a high degree of standardization.

One aspect of the training program's overall success is attrition. It has yet to be fully assessed. That assessment will not be available until summer's end. At that time users, developers and program managers will meet to evaluate training system effectiveness and to discuss any modifications.

TIME-COMPRESSED TANK GUNNERY TRAINING IN THE ARMY NATIONAL GUARD

Joseph D. Hagman
U.S. Army Research Institute
Alexandria, Virginia

John E. Morrison
Human Resources Research Organization
Alexandria, Virginia

Charles P. Lambert
Advanced Research Projects Agency
Arlington, Virginia

ABSTRACT

A device-based strategy is proposed for reducing or compressing the training time required to prepare Army National Guard armor tank crews for intermediate-level gunnery qualification on Table VIII. Using two computer-based devices, that is, the Conduct-of-Fire Trainer (COFT) and Guard Unit Armory Device Full-Crew Interactive Simulation Trainer - Armor (GUARDFIST I), time compression is accomplished in three ways. First, only Table VIII-related skills are trained on the devices. Second, emphasis is placed on training those Table VIII engagements typically not performed to standard. And third, training time is allocated primarily to crews that need it most, as determined through device-based competency pretesting. The strategy is designed for company-level implementation over three consecutive inactive duty training weekends.

ABOUT THE AUTHORS

Joseph D. Hagman is a senior research psychologist at the Army Research Institute's Reserve Component Training Research Unit in Boise, Idaho. He holds a Ph.D. in engineering psychology from New Mexico State University. His research interests are in soldier performance on armor simulation and training devices. Address: U.S. Army Research Institute, Reserve Component Training Research Unit, 1910 University Drive, Boise, ID 83725-1140. Phone: (208) 334-9390, FAX (208) 334-9394.

John E. Morrison is a senior staff scientist at the Human Resources Research Organization (HumRRO) office at Fort Knox, Kentucky. He holds a Ph.D. in experimental psychology from Tulane University. His research interests are in applying theories of learning and memory to problems in training.

LTC Charles P. Lambert is the Collective Training team chief for the Advanced Research Projects Agency's SIMITAR Project. He holds a Ph.D. in educational psychology from the University of Utah. His interests are in device-based training strategies for the Army National Guard.

TIME-COMPRESSED TANK GUNNERY TRAINING IN THE ARMY NATIONAL GUARD

Joseph D. Hagman
U.S. Army Research Institute
Alexandria, Virginia

John E. Morrison
Human Resources Research Organization
Alexandria, Virginia

Charles P. Lambert
Advanced Research Projects Agency
Arlington, Virginia

INTRODUCTION

In attempting to attain and maintain readiness standards comparable to their Active Component (AC) counterparts, U.S. Army National Guard (ARNG) combat arms units face significant training challenges stemming from imposed limitations on training time, that is, 12 Inactive Duty Training (IDT) weekends and a 2-week Annual Training (AT) period per year (U.S. Army Training Board, 1987). To make more efficient use of available tank gunnery training time, for example, ARNG armor units are seeking to shift from tank-based to device-based training. To maximize the payoff from such an approach, an effective and efficient strategy is needed to provide guidance on the design and execution of device-based gunnery training at the company level.

Previous tank gunnery training strategies designed for this purpose have either not provided enough specific "how to" guidance to support unit-level implementation (Headquarters, U.S. Army Training and Doctrine Command, 1992), failed to promote efficiency by requiring a full training calendar year to implement (Morrison, Campshure, & Doyle, 1991), or not emphasized device usage (U.S. Army Armor School, 1993). In contrast, the strategy proposed herein promotes efficiency through the maximal use of time-compression techniques and computer-based training devices. Its specific purpose is to prepare tank crews for successful first-run, live-fire gunnery qualification on Tank Table VIII (Department of the Army, 1993).

Two devices are used in the strategy: The Conduct-of-Fire Trainer (COFT) and the Guard Unit Armory Device Full-Crew Interactive Simulation trainer - Armor (GUARDFIST I). Time is compressed by (a) training only those skills and knowledges needed for successful Table VIII performance, (b) placing the focus of training on the most difficult Table VIII engagements, and (c) allocating training time primarily to crews that need it most.

APPROACH

The process of strategy development required (a) identifying the performance requirements of Table VIII, (b) determining the capabilities of the COFT and GUARDFIST I to support these training requirements, and (c) selecting the most efficient training approach for promoting the acquisition of gunnery skills on the devices as well as the transfer of these skills to performance on the tank.

Table VIII Performance Requirements

Table VIII consists of 12 tank gunnery engagements (see Table 1) divided equally into two groups of six engagements each: Table VIIIA is fired during the day; Table VIIIB is fired at night. Of these 12 engagements, two (A5S and B1S) are "swing" engagements that may be fired day or night, and two (A5A and B5A) are alternate engagements that may be fired in place of A5S and B5. Thus, each crew fires only 10 of the 12 engagements.

To promote transfer to live fire, device-based training should cover the actual engagements fired on Table VIII. For the sake of efficiency, however, training on all 12 Table VIII engagements may not be necessary if a subset of unique engagements can be identified that adequately covers the entire array of Table VIII tasks and conditions. To this end, two methods were used to reduce the number of engagements for training purposes. First, Table VIII engagement requirements were examined to identify duplication and to combine engagements accordingly. Second, engagements were ranked on difficulty of performance using data provided by Hagman (in press). As a result, eight unique engagements were identified and placed into three difficulty categories, as shown in Column 1 of Table 2. Shown in Column 2 are the specific Table VIII engagements covered by each unique engagement associated with the three difficulty categories.

Table 1
Table VIII Engagements

Table VIII (Day)

Engagement	Description
A1	On defense, engage a moving and a stationary tank with the main gun using the gunner's auxiliary sight (GAS) and battlesight gunnery.
A2	On defense, simultaneously engage a stationary BMP (tracked armored personnel carrier [APC]) with the main gun and a stationary BTR (wheeled APC) with the tank commander's (TC's) Caliber .50 machine gun.
A3	On offense, engage two sets of troops with the coaxial machine gun using precision gunnery.
A4	On offense and under nuclear, biological, and chemical (NBC) protection status, engage two stationary tanks with the main gun using precision gunnery.
A5A	On offense, engage a stationary and a moving tank with the main gun using precision gunnery.
A5S	On offense, engage two moving tanks with the main gun using precision gunnery.
	Table VIIIB (Night)
B1S	On defense, engage a stationary tank with the main gun from a three-man crew configuration using precision gunnery.
B2	On defense, engage two stationary BMPs with the main gun using precision gunnery.
B3	On offense and under NBC protection status, engage a stationary BMP with the main gun and a stationary RPG team with the coaxial machine gun using precision gunnery.
B4	On offense, engage a stationary and moving tank with the main gun using precision gunnery.
B5	On defense, engage a stationary tank with the main gun using GAS battlesight gunnery under external illumination.
B5A	On defense, engage a moving tank with the main gun using precision gunnery.

Difficult Engagements. Three of the four "difficult" engagements require employment of either the coaxial or Caliber .50 machine gun, either alone or in combination with the main gun. The fourth engagement of this category requires engagement of a moving target using the GAS. Besides being difficult to perform, these four engagements encompass most of the tasks and conditions encountered in Table VIII. Because these engagements are both difficult and comprehensive, they are the primary training objectives of the proposed strategy.

Fundamental Engagements. The next two engagements are called "fundamental" because they require crews to engage tank targets on either the offense or the defense without significant complicating conditions. Typically, these engagements are performed relatively well (Hagman, in press), presumably because they are not complicated by "additional" requirements such as using multiple weapon systems or engaging non-tank targets. Although the tasks and conditions of these fundamental engagements are encountered while practicing the difficult engagements, it is assumed that less proficient crews will learn these skills more efficiently under the simpler conditions of the fundamental engagements.

Special Engagements. The two engagements in this category are called "special" because they should be trained

Table 2

Difficulty Categories of the eight Unique Table VIII Gunnery Engagements and Associated Device-Based Training Exercises

Engagement Category		Device Training Exercises	
Description	Table VIII Engagements	COFT	GUARD-FIST I
Difficult Engagements			
On defense, engage simultaneous targets with the main gun and the TC's Caliber .50 machine gun.	A2	101 111[b]	---[a]
On offense, engage two sets of troops with the coaxial machine gun using precision gunnery.	A3	102 106	6A2
On offense, under NBC conditions, engage a stationary BMP with the main gun and an RPG team with the coaxial machine gun.	B3	101	6B3
On defense, engage a stationary and a moving tank target with the main gun using battlesight gunnery and the GAS.	A1, B5	113 117	6A1
Fundamental Engagements			
On offense, engage stationary or moving tank targets with the main gun using precision gunnery.	A4, A5S, A5A, B4	102 106 110	6A3 6A3 6A5 6B4
On defense, engage a moving tank with the main gun using precision gunnery.	B5A	105	6B5
Special Engagements			
On defense, engage two stationary BMPs with the main gun using precision gunnery.	B2	105	6B2
On defense, engage a stationary tank target with a three-man crew using precision gunnery.	B1S	103 107 119	6B1

[a]GUARDFIST I does not simulate the Caliber .50 machine gun and therefore is unable to support training on this engagement.
[b]COFT provides only part-task training for the TC on the Caliber .50 machine gun.

only under special circumstances, e.g., if the loader is inexperienced in changing from battlecarry SABOT ammunition to the HEAT rounds used for engaging lightly-armored vehicles, or if the TC is relatively inexperienced as a gunner under a three-man crew configuration. Typically, these special engagements are performed very well (Hagman, in press) and should not require training unless the above circumstances exist.

Device Capabilities

Regarding fidelity of simulation, COFT and GUARDFIST I are roughly comparable in that they both allow crews to use realistic tank controls in response to computer-generated images displayed through tank optics. They do differ, however, in certain respects. COFT, for example, is a stand-alone device that supports the training of only the gunner and TC, with inputs from the loader and driver simulated by an instructor/operator (I/O). In contrast, GUARDFIST I is a tank-appended device that supports the training of all four crew members, although the loader and driver simulation is at a lower level of fidelity than the TC and gunner simulation. In addition, COFT simulates all three M1 tank weapon systems (main gun, coaxial machine gun, and Caliber .50 machine gun), whereas GUARDFIST I simulates all but the TC's Caliber .50 machine gun, and therefore cannot support simultaneous engagement training.

Regarding training software, both devices offer evaluation exercises that present a heterogeneous set of engagements intended to simulate the array of Table VIII tasks and conditions, and training exercises that contain a more homogeneous set of engagements that focus on specific gunnery skills.

Evaluation Exercises. On COFT, the evaluation exercises (termed "gate" exercises) make up the last set of exercises in Group I of the recently fielded Advanced Matrix (U.S. Army Armor School, 1991). Each of the 16 gate exercises presents a different selection and ordering of 10 Table VIII engagements. On GUARDFIST I, a single Table VIII evaluation exercise, covering 10 of 12 Table VIII engagements, is provided as part of the final group of exercises (Group 6) in the GUARDFIST I training matrix (Industrial Data Link and Computer Sciences Corporation, 1994, February). Excluded from this exercise are the Simultaneous Engagement A2 (i.e., GUARDFIST I does not simulate the TC's Caliber .50 machine gun) and Engagement B5 (i.e., GUARDFIST I does not simulate external illumination for nighttime engagements).

Training Exercises. To support training, specific exercises were selected from Group 1 of the COFT advanced training matrix and Group 6 of the GUARDFIST I training matrix. To promote transfer to live fire, these exercises were selected to correspond closely to the eight unique Table VIII engagements. The two right-hand columns of Table 2 show the specific exercises selected for training purposes.

TRAINING APPROACH

The final step taken to support strategy development was to specify how training should be conducted in order to ensure maximum efficiency and effectiveness. The traditional bottom-up approach, where each crew begins training on easy engagements and proceeds to more difficult ones as proficiency increases, was judged to be inappropriate because of the limited amount of device-based training time available to ARNG tank crews. Thus, an alternative top-down approach was adopted where proficiency is first assessed and then followed by training at the highest engagement difficulty level indicated.

THE STRATEGY

Based on this top-down approach, the training strategy depicted in Figure 1 was developed. The strategy begins with a device-based pretest using GUARDFIST I or COFT to assess the need for device-based training. On GUARDFIST I, the pretest consists of Evaluation Exercises 6E1 and 6E2 which correspond to Parts A and B of Table VIII. These two exercises are to be administered four times, without feedback, so as to provide an adequate performance sample from which to make a valid assessment of crew proficiency (Smith & Hagman, 1992). On COFT, two exercises are to be selected from advanced matrix Gate Exercises 130-135 and two from Gate Exercises 136-139. Crews scoring 2800 points (i.e., 700 points per administration) or more are deemed "qualified" and not in need of device-based training. Crews scoring between 2800 and 1400 points on the pretest are deemed "partially trained" and would begin device-based training on the difficult engagements shown in Table 2. Crews scoring below 1400 points are deemed "untrained" and would begin training on the fundamental engagements (see Table 2) and then would proceed to the difficult engagements as proficiency dictates. For training purposes, proficiency is defined as destroying the targets on two consecutive attempts without committing a procedural error. Crews considered partially trained or untrained on the basis of their pretest performance may also receive training on the special engagements (see Table 2) if crew membership includes a new TC or loader. As a final step, all crews, including those considered to be qualified, would undergo a posttest, identical to the pretest, for the purposes of assessing their terminal device-based proficiency and of ensuring reliability of measurement. Under this strategy, it is anticipated that most crews will require less than 8 hrs of device-based training and testing, provided no special

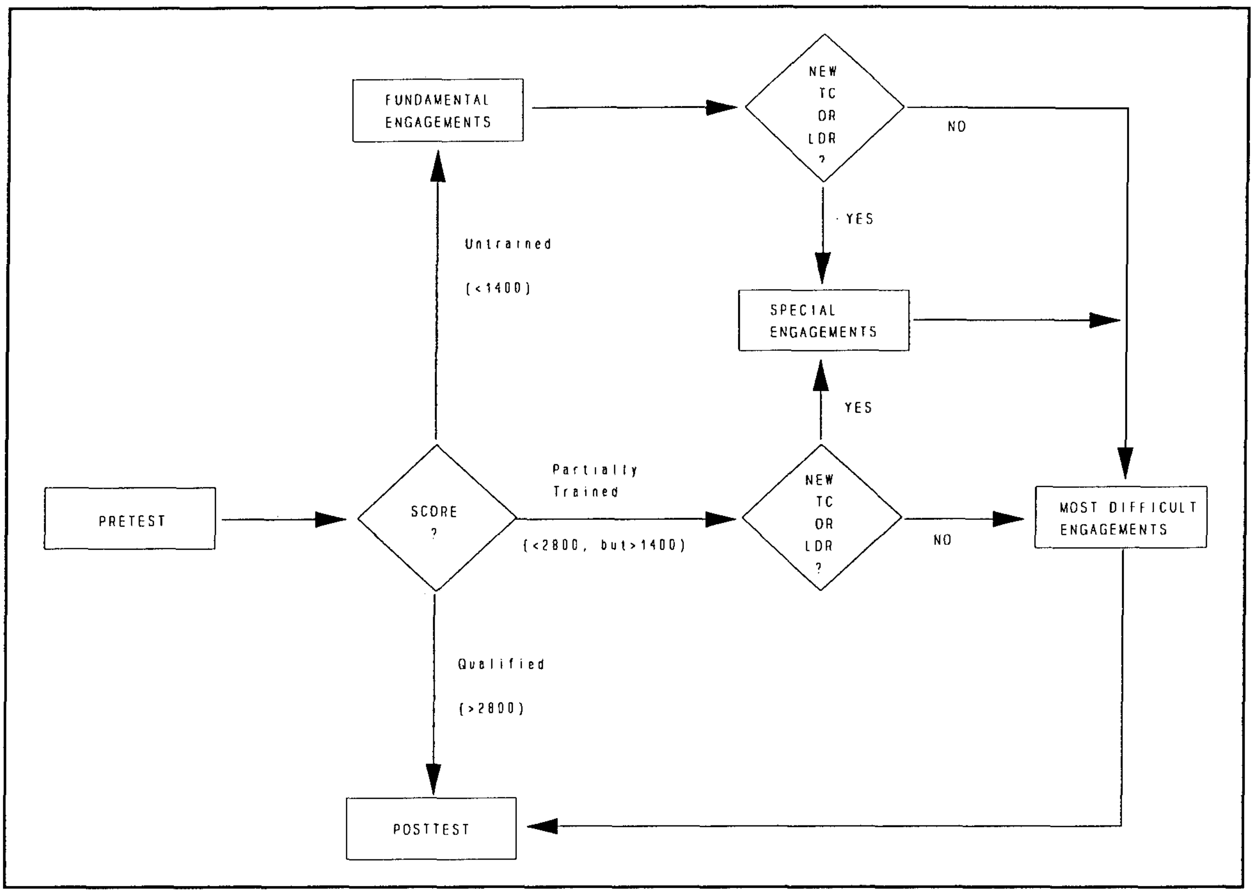


Figure 1. Flow of training and testing events in the strategy

engagement training is required (See Morrison & Hagman, inpress, for additional details.).

IMPLEMENTATION CONSIDERATIONS

The proposed strategy is designed to be implemented over three consecutive 3 IDT periods scheduled immediately prior to the AT period during which Table VIII will be fired. To maximize efficiency, device pretesting should be combined with Tank Crew Gunnery Skills Test (TCGST) administration. Pretesting should take about 60–90 min per crew.

Between the TCGST and the next IDT period, the unit commander and training noncommissioned officer (NCO) should review pretest performance to determine the appropriate training exercises for each crew. Depending on pretest performance, a crew may next undergo posttesting (i.e., qualified crews), training on the most difficult exercises (i.e., partially trained crews), or training on the fundamental exercises (i.e., untrained crews). Similarly, performance must be reviewed between the first and second and the second and third IDT periods, respectively. This review is required for selecting the appropriate training exercises and for determining which crews no longer require device-based training. The latter will ensure that limited available device time is diverted to crews that need it most.

If a unit has access to both COFT and GUARDFIST I, training should be scheduled such that crews practice the engagements on the device that provides the better simulation. The COFT, for instance, is the better device on which to train the simultaneous engagement, because only COFT provides a simulation of the TC's Caliber .50 machine gun. Once device-based training/testing is completed, tank crews should transition to the live-fire tank table exercises prescribed in FM 17-12-1-2 (Department of the Army, 1993).

In conclusion, several things should be noted regarding the proposed strategy. First, it does not provide sufficient training for loaders and drivers. Supplemental on-tank training will need to be scheduled, for instance, to give loaders practice at loading/unloading and drivers practice at starting/stopping and maintaining a steady gunnery platform.

Second, the strategy does not support the training of all aspects of tank gunnery. Skills and knowledges involving fire control system calibration, conduct-of-fire commands, and misfire procedures, for example, should be mastered either before or in conjunction with training on the devices. Development of a workbook is underway to support a concurrent approach to training these skills and knowledges (Pope, in preparation).

And lastly, in order to save time the proposed strategy recommends that COFT and GUARDFIST I be used in ways for which they were not originally intended. Thus, a formative evaluation is needed to test the validity of this approach in an actual ARNG setting. This evaluation should focus on determining whether or not (a) the devices are capable of providing efficient and effective training on the identified engagements, (b) a typical ARNG armor company can complete the recommended training in the time allotted, and (c) device-based training on the recommended engagements improves performance on Table VIII.

REFERENCES

Department of the Army (1993). *Tank Gunnery Training (Abrams)* (FM 17-12-1-2). Washington, DC: Author.

Hagman, J. D. (in publication). Performance analysis of Table *VIII tank gunnery engagements* (Research Rep.). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Headquarters, U.S. Army training and Doctrine Command (1992 August 13). *Combined arms training strategy development* (Draft TRADOC Pamphlet 350-XX). Fort Monroe, VA: Author.

Industrial Data Link Corporation and Computer Sciences Corporation (1994, February). *GUARDFIST I exercise summary.* Unpublished manuscript.

Morrison, J. E., Campshure, D. A., & Doyle, E. L. (1991). *A device/aid-based strategy for training M1 tank gunnery in the Army National Guard* (Research Rep. 1587). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Morrison, J. E., & Hagman, J. D. (in press). *A device-based, time-compressed strategy for Army National Guard tank gunnery training* (Research Rep.). Alexandria, VA: U . S . Army Research Institute for the Behavioral and Social Sciences.

Pope, G. E. (in preparation). *Tank gunnery workbook* (Research Product). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Smith, M. D., & Hagman, J. D. (1992). *Predicting Table VIII tank gunnery performance from M-COFT hit rate and demographic variables* (Tech. Rep. 955). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Smith, M. D., & Hagman, J. D. (in publication). *Predicting Table VIII tank gunnery performance from M-COFT hit rate* (Tech. Rep.). Alexandria, VA: U. S. Army Research Institute for the Behavioral and Social Sciences.

U.S. Army Armor Center (1991). *Instructor utilization handbook for the M1/M1A1 advanced matrix.* Fort Knox, KY: Author.

U.S. Army Armor School (1993). *Reserve Component Tank Gunnery Training Program* (ST 17-12-RC). Fort Knox, KY: Author.

U.S. Army Training Board. (1987). *Training and organization of the U.S. Army Reserve Components.* Fort Monroe, VA: Author.

DIGITAL VIDEO IN TRAINING

Dr. Ann E. Barron
University of South Florida
Tampa, Florida 33620

Susan Varnadoe
Analysis & Technology
Orlando, Florida 32826

ABSTRACT

The training industry is witnessing a transition from analog video stored on tape or videodisc to digital video stored on computer disks or CD-ROM. New compression techniques are making digital video technology more feasible for instructional applications such as interactive training, desktop video editing, and video conferencing.

There are several advantages to storing video in digital form. Digital video can be copied and reproduced without any loss of quality; whereas, each time an analog format is duplicated, the quality decreases and the noise level (imperfections) increases. In addition, digital formats offer the potential for increased manipulation; the images can be repositioned, resized, and recolored by a computer. Video in digital formats is also easier to transmit over computer networks.

This presentation will provide an overview of various digitizing and compression techniques for video. In addition, digital technologies such as QuickTime, Video For Windows, and Digital Video Interactive will be outlined. Demonstrations of various compression techniques will be included, and guidelines will be provided for selecting and implementing digital video in training applications.

ABOUT THE AUTHORS

Dr. Ann E. Barron is an Associate Professor in Instructional Technology at the University of South Florida where she teaches graduate level courses in multimedia and instructional design. She is an internationally recognized leader and presenter in the field of interactive training, the co-author of a recent book, *Multimedia Technologies for Training: An Introduction*, and the Executive Editor of the *Journal of Interactive Instruction Development.*

Susan Varnadoe is a Program Manager for Multimedia Development at Analysis & Technology, Orlando, Florida. She has managed, designed, and developed numerous projects for the U. S. Navy, U. S. Army, the Royal Australian Navy, Lockheed Corporation, and several commercial clients. Susan is a doctoral candidate in the Instructional Technology program at the University of Central Florida.

DIGITAL VIDEO IN TRAINING

Dr. Ann E. Barron
University of South Florida
Tampa, Florida 33620

Susan Varnadoe
Analysis & Technology
Orlando, Florida 32826

INTRODUCTION

For many years, interactive video in industrial and military settings relied on videodisc technology. Videodiscs provide 30 minutes of full-motion, full-screen video with two audio tracks. The video images on videodiscs are stored in analog format -- just like videotapes. Although analog video offers realistic colors and efficient storage, it requires a videodisc player and video monitor as part of the delivery configuration.

Recent advances in large storage media and digital compression techniques have provided the potential to record, edit, and store video in a digital format. This paper outlines the advantages and disadvantages of digital video and presents an overview of digital compression techniques and procedures.

ADVANTAGES AND DISADVANTAGES OF DIGITAL VIDEO

There are several advantages to storing video in digital form, but there are also some limitations. This section outlines the features and restrictions of digital video for training applications.

Advantages

High Quality Duplication. Digital video can be copied and reproduced without any loss of quality; whereas, each time an analog format is duplicated, the quality decreases.

Manipulation by Computer. Digital formats offer the potential for increased manipulation by a computer. With desktop video editing software, the images can be repositioned, resized, and recolored.

Networkable. Video in digital formats is easier to transmit over computer networks. Video teleconferencing over LANs and digital phone lines is possible with digital video.

One Monitor. A major advantage of digital formats (versus analog videodisc format) is the hardware requirement. With digital formats, the computer monitor can display both video images and computer graphics.

Disadvantages

Large File Sizes. A major impediment to digital systems is that digital video requires an enormous amount of computer storage space. For example, when one videodisc frame is digitized, a file of roughly one megabyte is produced. Without compression, less than 30 seconds of motion video can be stored on a CD-ROM disc.

Slow Transfer Rates. Another limitation to digital storage is the relatively slow data transfer rates of CD-ROM technology. At current transfer rates, it is difficult to display full-motion video at 30 frames per second in digital form.

Decreased Quality. The quality of the majority of the digital video currently available is less than the quality provided by VHS tapes.

COMPRESSION TECHNIQUES

In order to reduce the size of the digital video files, a variety of compression techniques can be used. General-purpose data compression programs, such as StuffIt, DiskDoubler, or PKZip have been around for years. These programs are used to compact computer programs and files for transfer or storage. For example, most of the files on the Internet are compressed by general-purpose programs, and many of the commercial software programs are distributed in compressed form because they require less diskettes. General-purpose compression programs can be referred to as "lossless." This means that when the file is decompressed it will be the exact size and hold the exact information as the original file.

When dealing with video compression, lossless compression is not powerful enough because they only allow you to reduce images at ratio of less than 4:1 (Anson, 1993b; Guglielmo, 1993). "Due to the huge size of digital image files, much larger compression ratios are needed [for video] and so most research into image compression has been applied to lossy compression schemes" (Anson, 1993a, p. 18). With lossy compression, some data is thrown away when the files are compressed. However, the lossy algorithms are "designed in such a way that it is difficult to see the loss, or at least so that crucial information is not lost" (Stern & Lettieri, 1994, p. 94).

Video compression techniques either employ spatial compression or temporal compression. In spatial compression (also referred to as intraframe), redundant or extraneous information is discarded on each screen. For example, if a portion of the screen is the same color, an algorithm will contain the information for that area of the screen, rather than storing information for each pixel. Spatial compression works well for images and still frames; however, it limits the compression possible for motion video.

In temporal compression (also referred to as interframe), redundant information is eliminated *between* screens. In other words, if the background of the scene does not change, the computer would save the first frame in its entirety; then for the next few frames, it will save only the parts of the screen that change. Most of the compression for motion video employs temporal compression because substantial redundancy exists between most frames and the compression ratio can be much higher.

CODECs

Compression programs are called "codecs," which stands for compressor/decompressor. Several codecs have been developed in the past few years; the sheer number of new hardware and software products for compression of video "is almost intimidating" (Nelson, 1994, p. 56). Some of the products provide proprietary compression algorithms, while others are adapting to emerging industry-standard techniques (Child, 1993). Some of the more common codecs include JPEG, MPEG, and Fractals. The compression scheme you choose depends on the desired resolution of the image, the storage space available, and the processing speed of your computer (Guglielmo, 1993).

JPEG

The international standard created by the Joint Pictures Expert Group (JPEG) is a compression technique that utilizes spatial compression for still frames. "JPEG compresses images at ratios of approximately 20:1 without noticeable loss of quality" (Guglielmo, 1993, p. 30). Although is JPEG is excellent for individual images, it does not offer enough compression for motion video, in which there may be up to 30 images per second. Another limitation of JPEG is that it does not include a standard for compression of an audio track.

MPEG

The Moving Pictures Experts Group (MPEG) developed a non-proprietary standard for motion video compression. The MPEG standard uses temporal coding to eliminate redundancy between frames, along with spatial coding to compact the information inside individual frames. Through MPEG, video compression of up to 50:1 (twice that of JPEG) can be achieved without noticeable loss. A disadvantage of MPEG is that most of the frames cannot be accessed individually. The MPEG format has been endorsed as the standard for CD-video that can provide linear movies on CD-I discs, and a new standard (MPEG-2) is being developed for broadcast video (Baron, 1992).

Fractals

Fractal compression is a powerful new technique that is appearing in the field. With Fractal compression, mathematics are employed to consider the similarities of objects and images. For example, "a fractal compression routine might find a way to approximate the image of a leaf out of three smaller versions of itself, overlapped" (Warren, 1993, p. 5). The image can then be represented by linear equations that only require a few bytes of memory. Fractal compression offers the advantages of fast decompression speeds, resolution independence, and much higher compression ratios than the other techniques (Anson, 1993b).

DIGITIZING VIDEO

To develop digital movies, a video digitizing card must be installed in a computer. Creating, editing, and playing digital movies usually involves several different software programs, including video capture software, video editing software, and digital movie players.

Video Capture Software

The first step in creating a digital movie is to "capture" the video. Capturing refers to converting the video from an analog source into a digital computer file. This process requires that a video digitizing board be used. These boards must be purchased and installed, unless the computer already has digitizing capabilities. The conversion process makes it possible to use a video camera, videotape, videodisc, or broadcast television as an input device and to display the video on a standard computer monitor.

A software capture program is used to control the capturing process. Decisions on frames per second, color depth, display size, and compression program affect the file sizes and the playback quality of the digital video. For example, a designer could record video at 15 frames per second instead of 30 frames per second or chose to set the display size at 1/4 of the screen, rather than the full screen. Both of these approaches will result in smaller file sizes. In general, the video capture procedure includes these steps:

- Connect a video source to the digitizing card
- Open the video capture software program
- Click on the "Record" button to start the recording
- Click on the "Stop" button for the end point
- Click on "Play" to preview the video clip
- Change the start and stop points, if necessary
- Save the file and give it a name

Video Editing Software

Although a minimal amount of editing can be done with the video capture programs, there are several powerful programs that specialize in video editing. One such program is *Adobe Premier* by Adobe Systems Incorporated. Video editing software allows you to sequence video clips, add graphic pictures and animations, and incorporate sound and visual effects. For example, if you have two video clips, you can change their length and join them together with a transition by using video editing software.

Video editing software is becoming very sophisticated, and many companies are using it to edit videotapes as well as digital video. In some cases, it is taking the place of extremely complex analog video editing equipment.

Digital Movie Players

The editing programs can also play the movies; however, software that is designed to play digital movies is available. Most movie players utilize a standard controller. The controller has the following options:

- Set audio level
- Play movie
- Pause movie
- Step through movie
- Slider bar to select a particular part of the movie

Movie-Savvy Applications

Many applications, such as Word Perfect, Microsoft Word, and Filemaker Pro, can now recognize and import digital movies. For example, you could write a resume in Word and embed a short movie into the document. It is important to note that the movie is not actually embedded into the document -- only the "pointers" to the movie are there, and you need to place the movie file on the same diskette with the resume.

Most of the hypermedia authoring systems now recognize and play digital movies. Programs such as HyperCard, HyperStudio, and ToolBook can incorporate buttons that show, hide, or play movies, based on the user's actions.

DIGITAL VIDEO TECHNOLOGIES

Until recently, most of the digital video that played on computers required special hardware. For example, Digital Video Interactive (DVI) has been around for years, but a special DVI-capable board is necessary for both capture and playback of the video. Two software technologies, QuickTime by Apple Computer and Video for Windows by Microsoft, have provided desktop video without special hardware.

Digital Video Interactive (DVI)

Digital Video Interactive (DVI) is a technique to digitize and compress video, which can then be stored on a CD-ROM disc or hard drive. A special computer board, made by Intel Corporation, is used to decompress the video on a computer when it is played back. This technology allows up to 72 minutes of full motion/full screen video to be stored on a CD-ROM. That is a tremendous amount of motion video when compared to the 30-minute limit of a

12-inch videodisc. DVI can also store other digital information, such as audio, text, and graphics.

QuickTime

QuickTime is a new format that was developed by Apple Computer Company to enable Macintosh computers to compress and play digitized video movies. A digitizing board is required to capture video for a QuickTime movie, but any color-compatible Macintosh computer can play the movies without additional hardware. The video is automatically compressed when the movie is created and decompressed when it is played back. Because QuickTime is a recognized Macintosh file format, the movies can be pasted or imported into a variety of Macintosh applications such as work processors, spreadsheets, and HyperCard.

Most QuickTime movies currently play in a small window on the Macintosh monitor (about one-quarter of the screen or less). Although the size can be expanded, the speed of the movies decreases substantially when you do so. In most cases the movies play at about 15 frames per second. The actual rate (frames per second) depends on the speed of the computer. For example, a movie on a Macintosh LC will play at a slower rate than the same movie on a Quadra or similar high-end Macintosh. Several codecs are available in QuickTime; others can be added as they are developed.

Video for Windows

Video for Windows is Microsoft's answer to software-only digital video for computers (Beer, 1993). It requires at least a 386 computer, 4 MB of RAM, and a digital audio card that is compatible with Microsoft's multimedia extensions. Similar to QuickTime, Video for Windows offers an easy to use interface to edit and play video clips without additional hardware.

CONCLUSION

Desktop digital video offers great potential for interactive training and videoconferencing in industrial and military settings. For example, instead of producing a videodisc to illustrate the procedure for performing maintenance on a helicopter, the video can now be digitized, saved as a QuickTime movie, and stored on a CD-ROM. As compression techniques improve, computers become faster, and alternatives for storage improve, the trend toward digital video in training applications is likely to continue.

REFERENCES

Anson, L. F. (1993a). Image compression: Making multimedia publishing a reality. CD-ROM Professional, 6(5), 16-29.

Anson, L. F. (1993b, October). Fractal image compression. *Byte*, 195-202.

Baron, D. (1992). MPEG-1 and MPEG-2: What's the difference? *Digital Media*, 2(7), 20(2).

Barron, A. E., & Orwig, G. W. (1993). *New technologies for education: A beginner's guide*. Englewood, CO: Libraries Unlimited.

Beer, J. (1993). Video For Windows: Microsoft's latest multimedia winner. *CD-ROM Professional*, *6*(5), 44-46.

Blank, S. (1992). Video image manipulation with QuickTime and VideoSpigot. *Advanced Imaging* 7(2): 53-55.

Bonomi, M. (1991). Multimedia and CD-ROM: An overview of JPEG, MPEG and the future. *CD-ROM Professional*, *4*(6), 38-41.

Drucker, D. L., and Murie, M. D. (1992). *QuickTime handbook.* Carmel, IN: Hayden.

Guglielmo, C. (1993). Lossy compression standards have extra squeeze appeal: JPEG, MPEG and related technologies maximize graphics compression. *MacWeek*, *7*(24), 30.

Gupta, S. (1993). Letting the computer drive computer video solutions. Computer design,

Leeds, M. (1991). Image compression. *Video Times* (Spring): 66-70.

Nelson, L. J. (1994). The latest in compression hardware & software. *Advanced Imaging*, *9*(1), 56-60.

Nicolaisen, N. (1993). Sound and vision: Digital video hits the desktop. *Computer Shopper*, *13*(12), 614(5).

Reveaux, T. (1992). Digitizers bring desktop video on board. *NewMedia* (August): 14-17.

Reneaux, T. (1993). Videodiscs: A shrinking market. *NewMedia 3*(5), 44.

Stern, J. L., & Lettieri, R. A. (1994). *BMUG's quicker QuickTime* (2nd Ed.). Berkeley, CA: BMUG, Inc.

Warren, S. (1993). Compression engines. *PC World*, *11*(11), 66(6).

DIGITAL VIDEO FOR MULTIMEDIA, WHAT ARE THE ALTERNATIVES?

David J. Sykes, Ph.D.
Paul C. Swinscoe
Hughes Training, Inc.
Arlington, TX 76011
(817) 695-2000

ABSTRACT

Digital video is becoming a viable alternative to the analog videodisc for multimedia in Computer Based Training (CBT) and other applications. The benefits of digital video are lower cost delivery platform hardware and more efficient processes for production, distribution, and maintenance. Today, there is a wide variety of hardware and software products available to implement digital video for multimedia including PLV, DVI, RTV, Motion JPEG, MPEG, Indeo, Quicktime, Cinepak, Ultimotion, and others. There is a wide variation in the quality and cost of these alternative solutions. Consequently, multimedia content developers are faced with a confusing array of options when it comes to using digital video. The objective of this paper is to compare the available methods of providing digital video to facilitate selection of the best approach for a given application. The paper includes a tabulation of performance, quality, and cost parameters to enable making informed choices. The different techniques of compression / decompression are briefly described together with the hardware and / or software needed to implement them. Decompression by software is particularly attractive since it does not increase the cost of the delivery platform. Hardware to play back the compressed video is also becoming more affordable and is the preferred solution when full motion, full screen video is required. Networking of digital video is briefly covered. The impact of emerging standards on the development of future products is discussed.

ABOUT THE AUTHORS

David J. Sykes is a Senior Scientist in the Technology Office of Hughes Training, Inc. and has over 25 years of experience in the development of training devices. For the last 10 years, he has been working on a wide variety of visual simulation systems ranging from large scale real-time image generators to videodisc and CD-ROM applications. He holds a Ph.D. in Engineering from Arizona State University.

Paul C. Swinscoe is a Software Engineering Manager for Commercial / Industrial Products of Hughes Training, Inc. He has more than 15 years of experience in the development of software for training systems and is currently responsible for developing software for multimedia-based training systems. He holds a Bachelor's degree in Engineering from the University of Manchester, England.

DIGITAL VIDEO FOR MULTIMEDIA, WHAT ARE THE ALTERNATIVES?

David J. Sykes, Ph.D.
Paul C. Swinscoe
Hughes Training, Inc.
Arlington, TX 76011

INTRODUCTION

Video has been an important ingredient of Computer Based Training (CBT) for many years. Until recently, the analog videodisc has been the only practical method of delivering video in an interactive CBT environment. Although not yet a mature technology, digital video promises many advantages over the traditional analog medium. The biggest challenge is developing affordable methods of compressing video so it can be handled by personal computers. Once digitized and compressed, the video becomes just like any other computer file (albeit a large file). Manipulation, editing, storage, distribution, and archiving become much easier and more convenient. Another benefit is the reduced cost and footprint of the CBT delivery platform. However, the biggest potential benefit of digital video is the ability to network the system. Once all elements of the multimedia are in digital form, they can be transmitted over networks thus avoiding the handling, distribution, and storage of disks and tapes. On the downside, there is a substantial investment in new equipment and training of personnel. Outsourcing of the process can be considered as an alternative to building a new capability in-house.

The objectives of this paper are to discuss the factors involved in selecting a digital video solution, to describe the various implementation methods available, and to present guidelines for selecting the best alternative to meet the requirements of a given CBT application.

BASIC CHOICES

The basic choices involved in the selection of video for a CBT system are shown in figure 1. The first choice is whether to use analog or digital video.

There are two reasons why the analog path could be chosen. The first is that there is already a large installed base of older PCs equipped with analog videodiscs. In this case, it is likely that all the CBT workstations would have to be replaced with new equipment if a digital solution is selected. The second is that the developer is unwilling to make the necessary investment to transition into the digital video world.

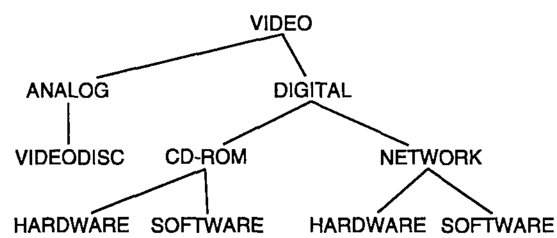


Figure 1. Basic Choices in Video for CBT

Assuming the digital approach is taken, there is the choice of CD-ROM or network. The CD-ROM is the prevalent method today but has many of the problems of the analog videodisc when it comes to mastering, handling, delivery, and updates to the content. Networking avoids all of these issues but requires an investment in servers and communications equipment.

In either case, CD-ROM or network, there is another choice to make. This is whether to use software or hardware for video decompression and playback in the CBT delivery stations. Each of these approaches can be implemented using a variety of products discussed later in this paper.

FACTORS TO CONSIDER

At this point, it is appropriate to discuss the factors to be considered when selecting a digital video implementation method. The following areas are used as the basis of a trade-off between alternative solutions:

- Image size and resolution
- Update rate
- Color resolution
- Qualitative aspects
- Accompanying audio
- Cost

Image Size and Resolution

There are three basic image sizes (shown in figure 2) although intermediate sizes are possible. Full screen is typically 640 (horizontal) x 480 (vertical) pixels, 1/4 screen is 320 x 240 and 1/16th screen is 160 x 120 pixels. Note that the resolution in pixels per inch of the display is the same in all three cases. The amount of processing required for a given resolution increases with the size of the image. For example, a full screen image of 640 x 480 requires 16 times as much processing as a 160 x 120 image. It is possible to configure the playback system to enlarge the image; for example, the 1/16th screen can be displayed at a quarter screen. The source resolution stays the same however, and a "blocky" or "mosaicked" image is the result. Also, frames may be skipped because of processor overload.

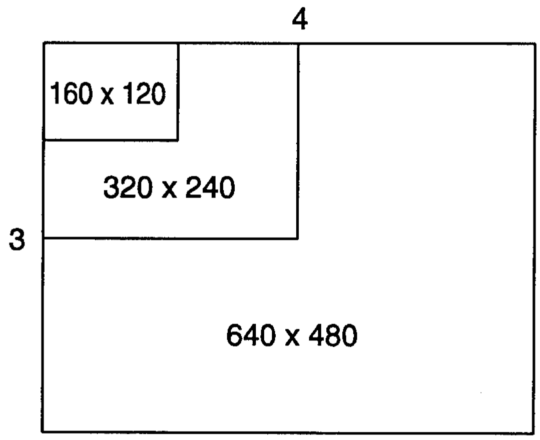


Figure 2. Image Size and Resolution

It is pertinent to consider the resolution of analog video to form a basis for comparison. The approximate indicators of resolution for some common video sources are:

Broadcast television	350 lines
VHS tape	250 lines
S-VHS tape	400 lines
Analog videodisc	450 lines

The term "lines" does not refer to the number of television scan lines, which is always 525 with 480 visible, it is the number of vertical lines that can be resolved in 3/4 of the screen width as shown in figure 3. Since the aspect ratio is 4:3, this number is equal to the perceived number of horizontal lines. Thus, VHS tape is roughly equivalent to a 320 x 240 digital image.

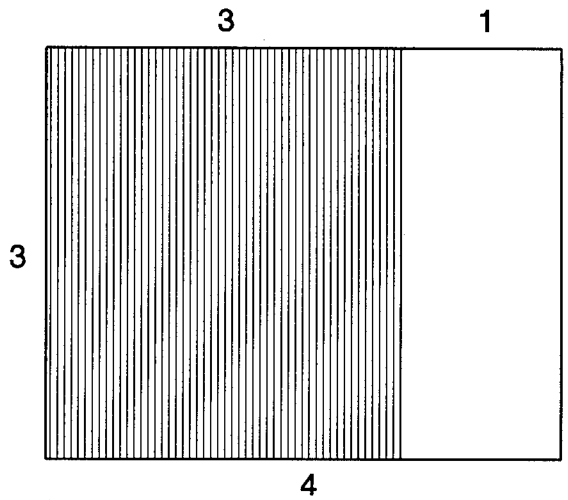


Figure 3. Measuring the Resolution of Analog Video

Update Rate

An update rate of 30 frames per second (fps) is often considered the norm. In some situations, a 15-fps rate may be acceptable and this reduces processing demand by a factor of 2:1. Actually, a 24-fps update rate is the minimum necessary for the eye to perceive continuous motion. This is the rate used in motion pictures.

Color

The use of 24-bit RGB can produce 16 million different colors. In the case of 16 bits the number of colors becomes 32,000 (approximately equivalent to NTSC video). The difference between 32,000 and 16 million colors is virtually imperceptible. If 8 bits are used however, only 256 colors are possible, which is marginal for most applications. Digital video compression schemes use techniques to provide adequate color with little extra processing compared to black and white.

Qualitative Aspects

In addition to the quantifiable factors described, there are qualitative aspects that must be considered. These are subjective in nature and the only way to judge them is to view a representative sample of the video. Compression of the video can reduce sharpness and create a fuzzy "washed-out" appearance. The scene content makes a difference. For instance, a close-up of a person's face looks better than a distant shot of a crowd of people or a group of buildings. Other artifacts that can occur are blockiness, aliasing (jagged edges), color streaking, scintillation, breakup, jumpiness, and torn edges.

Accompanying Audio

Since most video clips will need accompanying audio, it is important to consider the audio capabilities provided by a particular digital video implementation method. First, the various options for audio must be supported; second, the synchronization of the audio with the video must be acceptable. In particular, lip synchronization when a person is talking must look natural. The subject of audio is discussed later in the paper.

Cost

In the end, the choice of a digital video system will depend on its affordability. There is a trade-off between cost and overall quality. Figure 4 illustrates the general relationship between cost and overall quality including resolution, update rate, etc. The cost increases rapidly as one strives for better and better quality. It is difficult to put actual numbers on this curve since costs are constantly decreasing and quality is hard to define.

VIDEO FOR CBT

The use of video in CBT is focused around the type of training required. Different levels of quality, and therefore cost, can be applied to different CBT applications. There are four specific areas:

1. Training in equipment maintenance and operation. This requires detailed visual representations to show the intricacies of the machinery. Video in this case should be 1/4 screen or greater at a minimum of 24-fps.

2. Situational training such as sales, customer service and administrative duties. The video content is mainly of people rather than technical detail. For this application, the image size can be 1/4 screen or less at 15-fps or more.

3. Training with an on-screen instructor who guides the student through the course. Examples of this are training in computer skills, such as spread sheets and word processing. In this case the video can be 1/16th screen at 15- fps since most of the content will be text and graphics.

4. Training or education in which the student is drawn into the course to capture the person's attention. For this application, full screen at 24 to 30 fps is desirable.

In any event, it is critical that the customer and/or user of the CBT system see samples of the video early in the project. Customer acceptance of the video quality to be provided must be obtained before implementation begins.

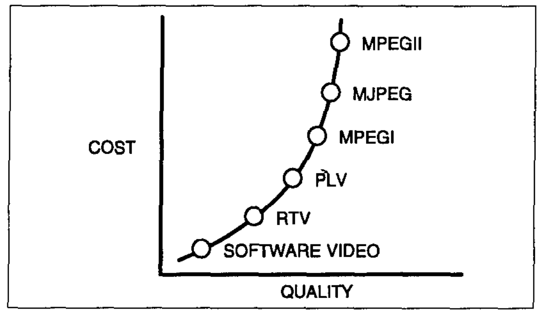


Figure 4. Cost of Digital Video vs. Quality

NEED FOR COMPRESSION

When full screen, full motion, full color video is digitized, the resulting data rate is 640 x 480 x 30 x 3 = 27 MB/second. This creates two major problems. First, a CD-ROM with a 650 MB capacity could only store 24 seconds of video. Second, no disk, CPU, computer bus, or network could transfer the data. Consequently, some way of compressing the data must be found. Many video compression schemes for PCs are designed to achieve a data rate of 150 KB/ second which is consistent with a single-speed CD-ROM. Going from 27 MB/second to 150 KB/second requires a compression ratio of 180:1. This means that 1 minute of video requires only 9 MB of storage instead of 1.6 GB.

No single approach is adequate to achieve such a high ratio, and therefore, a combination of several techniques is necessary [Ang 1991, Doyle 1994]. Compression methods rely on both redundancies in the data and non-linearities in human vision. Lossless methods such as run length encoding can achieve no more than a 3:1 ratio, so lossy techniques must be used. The lossy algorithms exploit aspects of the human vision system in such a way that information can be removed during compression without being noticed on playback. For instance, the eye is less sensitive to energy with high spatial frequency than energy with low spatial frequency. This deficiency is exploited by encoding the high frequency coefficients with less precision than the low frequency coefficients. Also, the eye is more receptive to detail in the luminance in an image than to the color. Consequently, the color can be sampled at a lower spatial resolution and with less precision than the luminance. Finally, video compression can take advantage of a feature in the video itself, namely the redundancy of information between adjacent frames. Many pixels do not change from one frame to the next. Also, groups of pixels forming objects move together from one frame to the next. These characteristics can be used to achieve major savings. Different compression algorithms employ some or all of the above techniques to varying degrees. The compression process is usually divided into two steps:

1. Intraframe Compression
   This step is applied to each frame individually. It consists of scaling and subsampling of the original image, color compression, and filtering of the higher spatial frequency components.

2. Interframe Compression
   This step is applied to a sequence of approximately 12 frames that have been intraframe compressed. One frame is designated a key frame, the remaining frames (called delta frames) are represented by differences from the key frame. If there is no difference, the delta frame is a "black" frame. Some interframe compression systems are very sophisticated and employ a variety of techniques such as bidirectional interpolation, and motion estimation and prediction. As a result, the process is computationally intensive.

Compression and decompression are performed by executing a pair of complementary algorithms. The decompression algorithm simply reverses the process of compression. A given compression / decompression algorithm pair may be implemented in several different ways with varying end results.

IMPLEMENTATION METHODS

The compression/decompression processes are performed by a **codec** which executes two complementary algorithms. The codec consists of two parts that can be combined into one package or the two parts may be completely separate. The various possibilities for design of a codec are as follows:

Compression	Decompression
Software	Software
Software	Hardware
Hardware	Hardware
Hardware	Software

Compression by software is performed as shown in figure 5. The video source is sampled by a capture board and the data is compressed in the CPU by a software codec.

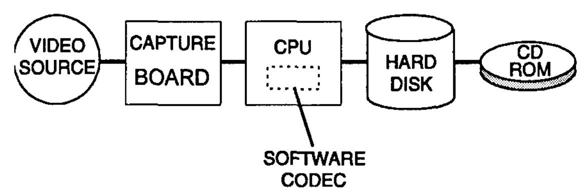


Figure 5. Video Compression by Software

The compressed file is stored on the hard disk where later it can be played back, transferred to a CD-ROM, or transmitted over a network. Compression by software usually requires much more time than is needed to decompress and is therefore termed an **asymmetric** process. The asymmetry ratio can be as high as 150:1 and depends on the type of CPU in use, the complexity of the algorithm, and the options selected. The CPU can be a PC, a workstation, or a mainframe. In general, the more time and resources expended during compression, the better the video quality on playback.

Compression by hardware is performed as shown in figure 6. The capture board includes the compression portion of the codec which does most of the work and requires little help from the CPU. The compressed data is sent directly from the capture board to the hard disk via the computer internal bus. The main benefit of hardware encoding is to achieve a symmetric system. This means that video can be captured and compressed at the same rate it will be played back; e.g., 30-fps, 15-fps. More data on video capture can be found in [Goodman 1994].

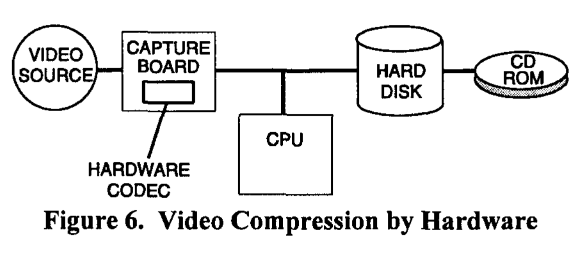


Figure 6. Video Compression by Hardware

Decompression by software is accomplished as shown in figure 7. The real benefit of this method is that no special hardware is necessary. However, the type and speed of the CPU is very significant. The difference in performance between a 386 CPU and a Pentium could mean the difference between 1/16th screen at 15-fps and 1/4 screen at 30-fps. A major feature of software decompression is scalability or the ability to automatically adjust the video playback to match the performance of the system. The same compressed video file can be played back with different resolution and frame rates depending on the CPU in use.

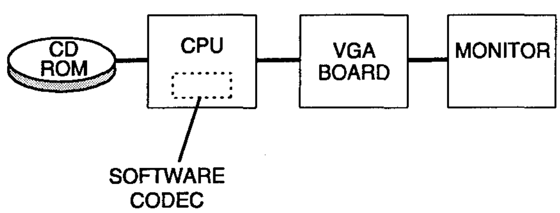


Figure 7. Video Decompression by Software

Decompression by hardware is depicted in figure 8. The decompression half of the codec is located on the playback board, the output of which is fed to the VGA card that drives the monitor. This approach is necessary if full screen, full motion video is required. It should be noted that many of these products are intended for entertainment applications since they allow 1 hour of continuous video to be stored on a single CD-ROM. Consequently, they may not provide all the features needed for CBT platforms. Examples of features needed for CBT are graphics overlay, freeze frames, jump to a frame, fast play, and reverse play. These should be taken into account when selecting a video playback board.

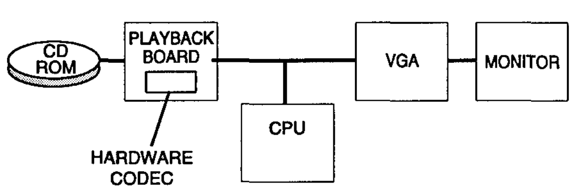


Figure 8. Video Decompression by Hardware

Software Hierarchy

A software hierarchy is necessary to control the video playback. This is true for both hardware and software codecs as shown in figure 9. A media player under the Operating System is responsible for controlling the video playback. The same media player also controls sound and animation. Device drivers are configured into the software system for each type of "device"; i.e., video codec, sound card, etc.

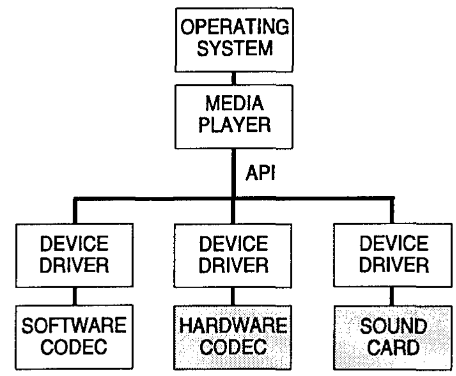


Figure 9. Software Hierarchy

The device drivers control the actual codecs whether they are hardware or software. Thus, in the case of a software codec there are four layers in the software hierarchy: OS, media player, device driver, and codec. The interface between the media player and device drivers is called the Application Program Interface (API). The most common API is the Microsoft Media Control Interface (MCI) which is supported by most suppliers of video products for the PC.

DIGITAL VIDEO STANDARDS AND PRODUCTS

So far, the discussion on digital video has been in general terms; now the characteristics of specific standards and products will be described. Further details can be found in [Sirota 1993, Ozer 1994, and Wallace 1994].

Indeo

Developed by Intel, Indeo is a codec that uses hardware for real-time compression and software for decompression. The image on playback is scaleable from 160 x 120 to 320 x 240 at 15 to 30-fps depending on the power of the CPU. Indeo is included in Microsoft Video for Windows Package.

Cinepak

This codec by SuperMac uses software for both compression and decompression. Consequently it is an asymmetric codec. Cinepak plays back images scaleable from 160 x 120 pixels to 320 x 240 pixels at 15 to 24-fps. Quicktime includes the Cinepak codec.

Ultimotion

IBM distributes Ultimotion as part of OS/2 2.1. It is an efficient derivative of Photomotion. It is a symmetric compression process using software only and the resolution is limited to 160 x 120 pixels at 15-fps.

DVI (Digital Video Interactive)

This is a hardware-based codec developed by Intel. The hardware is the Intel Action Media II board which was originally priced at $2,000 but is now much less expensive. DVI has two levels: presentation level video (PLV) and real-time video (RTV). PLV offers full-screen, 30-fps playback but compression must be done at a service bureau at a cost of $175 per finished minute. RTV can be compressed by the hardware codec at the PC level, but the quality is inferior to PLV.

True Motion

True Motion can play back full-screen video at 30 fps. The data rate is 600-KB/second which exceeds the data rate of single, double, and triple speed CD-ROM drives. Compression must be done at a service bureau at a cost of $200 per finished minute. Hardware is required for the playback and is the same as used for DVI.

Motion JPEG

This algorithm was developed by the Joint Photographic Experts Group,(JPEG). It was originally intended for still images but has been extended for video. It uses intraframe compression only and consequently has a low compression ratio and high data rate. Hardware is used for both compression and decompression in a symmetric process. A major limitation of Motion JPEG is that it does not support audio.

MPEG - 1

Of all the codecs available, those based on MPEG-1 show the most promise. MPEG-1 is an international standard developed by the Motion Pictures Experts Group, (MPEG). It includes both intraframe and interframe compression and was specifically designed for a data rate of 150 KB/second and is thus ideal for CD-ROMs. It is an open standard and any company can develop codecs based on it. Video encoded with MPEG can be played back on any device that is MPEG compatible. However, there are options during encoding that can vary the quality of the video on playback. MPEG-1 provides 352 x 240 pixel resolution at 30-fps and has interleaved audio. Hardware is necessary for playback and boards ranging in price from $400 to $800 are available.

Today, compression must be done by a service bureau at a cost of up to $150 per finished minute [Mills 1994]. Affordable hardware will be available for MPEG compression in the future [Magel 1994]. An MPEG video stream includes embedded timing data so the decoding process regulates playback at 150 KB/second. Consequently, increasing the speed of the CD-ROM drive or use of a hard drive does not improve the playback conditions. A compressed video file requires a 9 MB of storage for 1 minute of playback time.

MPEG-2

Whereas MPEG-1 was designed for CD-ROM-like devices, MPEG-2 is intended for broadcast television to replace NTSC. It supports a resolution of 720 x 480 pixels at 30-fps and consequently has a much higher data rate than MPEG-1. It is unlikely that MPEG-2 will be suitable for CBT applications in the near term. However, it does show promise for the future when networks of adequate bandwidth become available.

Comparison

The codecs and standards described are compared in table 1. The list is not exhaustive; there are other evolving approaches based on wavelets and fractals [Perey 1994].

Table 1. Comparison of Video Compression Methods

Method	Resolution	Frame Rate (FPS)	Data Rate (KB/s)	Audio	Hardware Playback	Compression Process	Quality
Indeo	160 x 120 320 x 240	15-30	150	Yes	No	Asymmetric/ Symmetric	Low-Med
Cinepak	160 x 120 320 x 240	15-24	150	Yes	No	Asymmetric	Low-Med
Ultimotion	160 x 120	15	150	Yes	No	Symmetric	Low
PLV	256 x 240	30	150	Yes	Yes	Asymmetric	High
RTV	128 x 120	30	150	Yes	Yes	Symmetric	Low
Motion JPEG	640 x 480	30	600-1500	No	Yes	Asymmetric	High
MPEG-1	352 x 240	30	150	Yes	Yes	Asymmetric	Med-High
MPEG-2	720 x 480	30	150-2000	Yes	Yes	Asymmetric	Very High
True Motion	640 x 480	30	600	Yes	Yes	Asymmetric	Very-High

ENHANCING DIGITAL VIDEO DURING CAPTURE

Unless prerecorded footage is being used, there are several techniques to use during capture to enhance the quality of the compressed video when it is eventually played back [Holsinger 1994].

1. Pay attention to proper lighting to minimize shadows.
2. Avoid black backgrounds.
3. Use close-ups rather than long shots.
4. Minimize the rate of change of scene content when panning or tilting.
5. If text is used, use large characters; font of size 10 or 12 points will not be legible after compression. It is better to avoid text in the video itself and to use graphics to superimpose the characters on the video during playback.

AUDIO

While audio does not present the same challenge as video, the subject deserves proper attention. There are three decisions to be made when capturing audio along with the video. These are:

Sampling Rate:	11, 22, or 44 KHz
Resolution:	8 or 16 bits
Presentation:	Mono or stereo

The choice of 8-bit, 11-KHz mono is adequate for narration with a male voice. This requires only 66 KB per minute which is small compared to 9 MB per minute for the video. At the other extreme 16-bit, 44-KHz stereo uses up 10.5 MB per minute; i.e., more than the compressed video. With audio compression this can be cut to about 2 MB per minute which is still significant. However, there is no need to use the highest quality audio option for CBT. Typically, 16-bit, 22-KHz mono is perfectly adequate for most CBT applications and results in less than 1 MB per minute if audio compression is used.

If MPEG is chosen as the video system, audio choices become easier because MPEG defines both video and audio compression standards. MPEG audio compressors start with 44-KHz sampled stereo sound and reduce it to less than 2 MB per minute without any discernible loss of quality. This ensures that the interleaved video and audio can be played back from a 150-KB/s CD.

The subject of audio/video synchronization should not be overlooked particularly when "talking heads" are part of the training scenario. Generally, the audio file is interleaved with the video file on a frame-by-frame basis. A good example is the Audio Video Interleaved (AVI) file used in Video for Windows. This approach does not guarantee synchronization of the final output however, because different delays may occur in audio processing and video processing. This is another reason to test the system using the identical hardware and software that comprise the CBT delivery station.

VIDEO NETWORKING

In the networked approach, a file server contains the video along with other multimedia components stored on a hard disk [Furcht 1994]. Because of the large file sizes, the disk capacity must be several gigabytes. This is not unreasonable however, since the cost of hard drives is now around $700 per gigabyte and falling. The CBT delivery stations are connected to the server by some type of Local Area Network (LAN) as shown in figure 10.

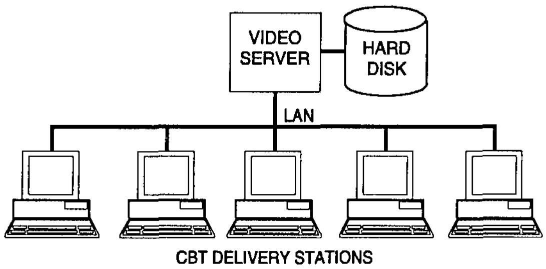


Figure 10. Transmitting Video Over a Network

There are two ways to consider transmitting video over a LAN. Th first way is to download the video in non-real time before the training session starts. The video then resides on the hard drive within each CBT workstation until new training material is required. This approach is perfectly feasible with networks such as Ethernet. Since the transfer is from hard drive to hard drive, the data rate can be much higher than CD-ROM speeds. Consequently, the time to down-load is much less than the normal video playing time. The second way of networking video is real-time on-demand transmission. Unfortunately, Ethernet was never designed to handle long uninterrupted streams of data with critical timing as in the case of digital video. Therefore, video on-demand via Ethernet is not recommended although it can be made to work in some circumstances. The implementation of digital video on demand needs network solutions such as FDDI (Fiber Distributed Data Interface) or ATM (Asynchronous Transfer Mode). Neither of these is very affordable today. In the future, ATM products will dominate the market for most networking applications. A comparison of LAN protocols is shown in table 2.

Table 2. LAN Protocols

Protocol	Bit Rate	Access Time
Ethernet	10 Mbps	Indeterminate
Token Ring	4-16 Mbps	50-500 ms
FDDI	100 Mbps	10-200 ms
Fast Ethernet	100 Mbps	Indeterminate
ATM	150 Mbps	Det by switch

When a fully networked solution is implemented all the problems with mastering, distributing, and handling CD-ROMs are eliminated. In particular, updates can be made quickly and easily.

SOME THINGS TO REMEMBER

1. Trade-off hardware versus software solutions. The cost of upgrading the CPU for a software algorithm may exceed the cost of adding video playback hardware. Consider these examples. For a 1/4 screen or less, use a software algorithm and a minimum processor of 486/DX2/66MHz. For greater than a 1/4 screen, use an MPEG playback card with a somewhat less powerful CPU.
2. Ensure the chosen solution for digital video has all the required features needed for CBT.
3. Due to the dynamic environment, buy the hardware and software as late as possible.
4. Stay with the industry standards including hardware and software interfaces to ease expansion and upgrade.
5. Consider any possible cross-platform requirements i.e. portability between Windows, System 7, Unix, and OS/2.
6. Test the whole system using the same hardware/software configuration as for the delivery platform.
7. Show samples of the video with the chosen implementation to the customer as early as possible.
8. Remember, the compression removes high spatial frequencies which blurs sharp edges and removes some detail.
9. Many small companies in digital video come and go, so make sure that your suppliers are likely to be around for awhile.
10. Be very careful when planning a network solution using digital video until the next generation of networking products is available.

FUTURE TRENDS

Digital video for CBT is still in its infancy and many improvements can be expected in the future. As price/performance of CPUs continues to improve over the years, it is expected that CPU performance will double every 2 years. The price per megabyte of hard drives will continue to fall. Hardware codecs will become inexpensive and become a standard part of computer graphics (VGA) boards. Many of the digital video products available today will disappear from the market. The most probable survivors will be MPEG and Indeo.

Digital video will be driven by the entertainment market but CBT developers will be able to take advantage of the situation. The CD-ROM will be overtaken by networking and (in due course) the ATM protocol will be the most attractive solution. One can expect a continuously improving capability over the next decade. Finally, never before in history has a new technology been so affordable.

CONCLUSIONS

Digital video has evolved to a point where it is possible to replace the analog videodisc with an all digital system. There are many varieties of digital video with a wide range of quality and cost; trade-offs are necessary to determine the optimum solution to a CBT problem. Quality will improve and costs will decrease over time. Therefore, the developer should buy the hardware and software as late as possible. It is very important to show samples of the video to the customer before committing to a solution. One should follow the standards and avoid the products based on proprietary algorithms and interfaces. This will avoid being locked into one manufacturer's product line and enable the developer to take advantage of new hardware and software. The entire CBT system should be tested using the exact platform configuration on which it is to be delivered to avoid any surprise bottlenecks. Finally, if the digital video world seems too much to handle, the possibility of outsourcing should be considered.

REFERENCES

- Ang, Peng H. (1991). Video Compression Makes Big Gain. *IEEE Spectrum*, October 1991, 16-19.
- Doyle, Bob (1994). How Codecs Work. *New Media Magazine*, March 1994, 52-55.
- Furcht, Borko (1994). Multimedia Systems: An Overview. *IEEE Multimedia*, Spring 1994, 47-59.
- Goodman, Ben (1994). Ready for Action: Five Video Capture Boards Bring Motion Video to Your PC. *Computer Shopper*, February 1994, 204-212.
- Holsinger, Erik (1994). Mastering Techniques for Working with MPEG. *Digital Video*, June 1994, 68-70.
- Magel, Mark (1994). MPEG Encoders. *Digital Video*, October 1994, 50-60.
- Mills, Karen (1994). How and Why to Use a Video Service Bureau. *Desktop Video World*, March 1994, 36-43.
- Ozer, Jan (1994). Digital Video: A Comparison of Codec Choices. *Multimedia Monitor,* July 1994, 14-17.
- Perey, Christine (1994). Creatures of Habit. *Desktop Video World*, March 1994, 28-34.
- Sirota, Warren (1993). Compression Engines. *PC World*, November 1993, 66-72.
- Wallace, Lou (1994). Compression/ Decompression. *Info World*, March 7, 1993, 82-83.

AN INTERACTIVE MULTIMEDIA TUTOR FOR SOFTWARE SYSTEM MAINTENANCE

Jean D. Garthwaite, George A. Huff
The MITRE Corporation
Bedford, Massachusetts

ABSTRACT

Eighty percent of the life-cycle cost of a software-intensive system is for maintenance, and the cost of fixing software increases by an order of magnitude as it is passed from developer to maintainer. Much of the requisite knowledge about the system's mission, software structure, and maintenance toolset and procedures resides with the developers of a software-intensive system. The software maintainers must painstakingly recreate it.

Current acquisition documentation and training approaches, such as those required by DOD-STD-2167A, do not effectively convey software system knowledge to the maintenance organizations. Typically, these approaches consist of one-time only system-specific training coupled with large volumes of procured documentation that provide information in a format that is more relevant to those acquiring and developing the system rather than those maintaining it. In general, maintainers find this training too brief and the documentation too hard to understand and use, so they resort to spending large amounts of time reading the source code to derive the information they need. To reduce the learning curve among software maintainers and the associated life-cycle costs for specific systems, new software maintenance training approaches must be developed that effectively capture and transfer this system-specific knowledge to the maintenance organizations.

This paper describes the effort to develop a new software maintenance approach, a computer-based tutor, for the Higher Authority Communication/Rapid Message Processing Element (HAC/RMPE). The HAC/RMPE tutor prototype, with its training system and on-line performance aid features, is a unique approach for supplementing the software maintenance training process. It captures the knowledge about the system mission, software structure and maintenance environment. Thus, it addresses the steep learning curve associated with a new maintenance trainee gaining proficiency with an unfamiliar project, software system, and maintenance environment. It also packages information in a format, hypermedia, that to date is atypical for software maintenance. The concept of a system-specific software maintenance tutor as exemplified in the HAC/RMPE prototype shows great promise for filling a critical gap in the training of software maintainers and thereby reducing the life-cycle cost of system maintenance

ABOUT THE AUTHORS

Jean Garthwaite is a Member of the Technical Staff at The MITRE Corporation. She serves as a technology expert for the Training Technology & Development Department and has extensive experience in large-scale software development. She holds a Bachelor of Science degree in Computer Science from the University of New Orleans and a Master of Science degree from the Center for Advanced Computer Studies at the University of Southwestern Louisiana.

George Huff is a lead engineer at The MITRE Corporation. His interests include formal methods of software specification and verification, software development process, metrics, software acquisition, and computer security. He is currently a member of MITRE's Training Specialty Group, where he leads the HAC/RMPE Software Maintenance development team. He has also been involved in several other training system prototype developments, and has developed and evaluated courseware authoring tools. He has mathematics degrees from Kenyon College and Lehigh University and a Ph.D. in philosophy from Stanford University.

AN INTERACTIVE MULTIMEDIA TUTOR FOR SOFTWARE SYSTEM MAINTENANCE

Jean D. Garthwaite, George A. Huff
The MITRE Corporation

INTRODUCTION

Cost Benefit of Improved Training

Eighty percent of the life-cycle cost of a software-intensive system is for maintenance [1], and the cost of fixing software increases by an order of magnitude as it is passed from developer to maintainer. This cost increase is not due to maintenance staff insufficiently trained in the principles of software development. Rather, it is due primarily to lack of adequate training of maintenance personnel in the system mission, the structure of the implemented software, and the suite of tools and procedures that comprise the maintenance environment [2].

Current acquisition documentation and training approaches, such as those required per DOD-STD-2167A [3], do not effectively convey this knowledge to the maintenance organizations. Typically, these approaches consist of one-time only system-specific training coupled with large volumes of Software Requirements Specifications, Software Design Documents, etc., that provide information in a format that is more relevant to those acquiring and developing the system rather than those maintaining it. For example, the traditionally delivered text-based documentation provides a functional orientation of the software as opposed to a thread orientation, such as a maintainer would need to know to understand how the software is crafted. In general, maintainers find this training too brief and the documentation too hard to understand and use, so they resort to spending large amounts of time reading the source code to derive what is the software architecture, what is the mission of the system, etc.

Much of the requisite knowledge about the system's mission, software structure, and maintenance toolset and procedures resides with the developers of a software-intensive system. The software maintainers must painstakingly recreate it. Therefore, to reduce the learning curve among software maintainers and the associated life-cycle costs for specific systems, new software maintenance training approaches must be developed that effectively capture and transfer this system-specific knowledge to the maintenance organizations.

Tutor Concept Applied to Software Maintenance

One proposed solution to this maintenance training need is the concept of a computer-based software maintenance tutor that in a single form captures and presents specific knowledge about a system's mission, software structure, maintenance toolset and supporting procedures. To investigate this concept, we have prototyped a software maintenance tutor for a specific program. The program is the Higher Authority Communication Rapid Message Processing Element (HAC/RMPE) subsystem of the Rapid Execution and Combat Targeting (REACT) system. REACT is designed to integrate and update the Intercontinental Ballistic Missile (ICBM) system in the areas of Higher Authority message processing and weapons systems command and control. The HAC/RMPE component is a system implemented in Ada that integrates several communications channels, performs rapid message processing, alarm integration, and delivers error-corrected messages to the missile crew.

Program Background

The HAC/RMPE maintenance concept is to use an organic maintenance organization, known as the HAC/RMPE Software Support Facility (HSSF), to perform periodic updates to the message processing software, to make minor bug fixes, and to perform system enhancements, as required. This facility is to be staffed with both civilian and military personnel. Because a high turnover rate is anticipated with military personnel, training is needed to indoctrinate new programmers quickly on the HAC/RMPE mission, software architecture, and maintenance procedures and environment. In addition, given the complexity of the software and the expected rate of change, training is needed to enable experienced personnel to refresh their knowledge or to access quickly high-level information about the software architecture.

Scope

This paper describes the overall concept of a software maintenance tutor as reflected in the HAC/RMPE prototype. In particular, we describe the objectives and presentation techniques of the HAC/RMPE prototype, the design approach, and lessons learned to date.

TUTOR OBJECTIVES and PRESENTATION TECHNIQUES

Training Objectives

To determine the training objectives for a computer-based software maintenance tutor for HAC/RMPE, we conducted a training needs assessment that included detailed discussions over many months with the tutor's intended users, developers, project office management, and HSSF management. The resulting objectives, identified below, reflect the tutor's primary function as an accessible repository for technical and procedural information:

1. Provide introductory material for new HSSF personnel. New personnel will not be familiar with the specifics of the HAC/RMPE software they are to maintain, nor will they have experience with the administrative and technical procedures established for the HSSF, in such areas as configuration management, security, and qualification. However, new HSSF personnel are expected to be familiar with the Ada programming language, or at least they will be trained in Ada through other means than the tutor.

2. Capture and link important system, mission, and development knowledge that's not contained in the documentation traditionally delivered with the system. Examples include design rationale and trade-offs, and details of the software tasking architecture.

3. Present materials in ways that cross technical or documentation boundaries, rather than duplicating material pertaining to a single view. For example, present a documentation overview that covers the interrelationships among the various documents to demonstrate the concept of requirements traceability and qualification from the System Operational Requirements Document to the source code modules.

4. Provide an on-line reference aid for more experienced personnel, following a "presentation-oriented" style of instruction. Do not provide interactive problems where the user is required to "guess" the correct answer, since there is often no one "right" answer for implementing a software modification.

These objectives are not directed towards a structured sequence of instructional "modules" that trainees are expected to master before continuing on to actual maintenance activities. Thus, much of the traditional functionality of a computer-based training system -- curriculum management, student assessment, coaching, and so on -- is not required in the tutor. Rather, these objectives are clearly directed toward effective and efficient retrieval of technical materials, and integration of these materials.

Presentation Strategies

Given the training objectives for a HAC/RMPE software maintenance tutor that is an interactive browser and a reference aid and not a traditional training system, we analyzed presentation strategies to best meet the needs of such a tutor. The key to meeting these training objectives lies in utilizing powerful and sophisticated presentation techniques that ensure the tutor materials are easily accessed and memorable. The resulting techniques selected for the prototype are as follows:

1. Use a textbook metaphor, with a browsable table of contents and material organized into chapters and subchapters. This is a comfortable metaphor for new users, and it brings with it implications about the order of topics.

2. Use graphic depiction wherever possible. Diagrams help the trainee visualize the topics being presented -- mission, software interfaces, communication channels, software architecture, and task structure -- and also to make hierarchical connections between topics. Furthermore, appropriate graphics are more memorable than text and function more intuitively as a navigation tool.

3. Provide immediate access to software structure information. Since every software change is made within a larger context, a trainee must be taught to refer to the software structure as a means of understanding the potential ripple-effects that a given change could have on the whole. In the DOD-STD-2167A context [3], this means the static structure of the software, in terms of the decomposition into Computer Software Configuration Items (CSCIs), Computer Software Components (CSCs), and Computer Software Units (CSUs), must be presented and reinforced.

4. Present introductory materials in a tour format, so that the user can view all the contents, from highest level to most detailed, by simply clicking a *Continue* button.

5. Provide integrated examples of completely worked out software changes, based on the kinds of changes that are anticipated for the HAC/RMPE software. Per the training objectives, these practice problems are to be presentation-oriented rather than interactive.

6. Present topics using hypertext and multimedia, as appropriate, to link information in a more readily accessible format, overcoming the bounds of traditional paper documentation.

7. Use short video clips to depict the software capability in the context of the overall military mission it supports.

HAC/RMPE TUTOR PROTOTYPE DESIGN

In accordance with the established training objectives and presentation strategies, we implemented a prototype of the HAC/RMPE software maintenance tutor. The resulting tutor framework, as applied to the HAC/RMPE problem domain, is comprised of the following components: a MainMenu Interface; a guided overview of REACT and HAC/RMPE; a guided overview of the HAC/RMPE software development environment; a HAC/RMPE documentation trace; and several hypertext chapters explaining key HAC/RMPE software components.

The material covered by the HAC/RMPE tutor prototype starts with the real world concept of the REACT mission including the source of incoming messages, the processing functions performed by the software, and the output and displays provided to the end user. This kind of information is provided in order to promote the understanding of the capability that the software indeed implements. Instilling a visual picture of the capability fosters the staff members' fluency with the application and helps develop the intuition needed for troubleshooting software fixes and making enhancements.

Graphical Menu Interface

The MainMenu Interface serves as an on-line table of contents for the HAC/RMPE tutor prototype. It provides the trainee with easy access to the array of program information contained in the other training components.

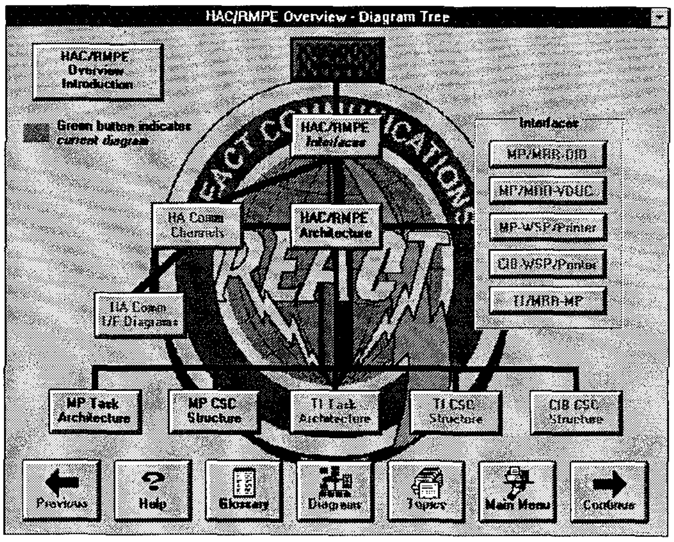


Figure 1. The Diagram Tree

Program Overview Component

The HAC/RMPE Overview uses a set of functional diagrams and block diagrams to guide the trainee through the basics of the REACT mission and the HAC/RMPE software system. Each diagram with its associated text, voice-overs, video, and graphics tells a story about some aspect of HAC/RMPE. A natural progression from the topmost level to the details is built in: the tour starts with the REACT and HAC/RMPE mission overview, and proceeds through external interfaces, hardware and software architecture, CSCI structure, CSC structure, and finally the Ada tasking structure. The diagrams help the trainee visualize the topics being presented--mission, software interfaces, communication channels, software architecture, and task structure. They also help to make hierarchical connections between topics. Furthermore, appropriate graphics and video add more realism to the subject matter. Figure 1 shows the "Diagram Tree" which functions as an index screen to the overview topics.

Equipment and Documentation Tour Components

The HSSF Equipment Tour uses a schematic of the HSSF development environment to introduce the trainee to the hardware, software, and compilation procedures used to generate a new release of the HAC/RMPE software. The trainee can take the guided introduction to the equipment schematic or inspect any piece of equipment in detail. Equipment details are provided in the form of pictures and a set of relevant textual topics. Figure 2 is a screen from the Equipment Tour showing one of the nine topics relating to the VAX 6320 development processor.

The Documentation Tour provides a quick overview of the various kinds of documents expected to be useful for maintenance. Most were created by the HAC/RMPE contractor as required by DOD-STD-2167A [3]. The tour starts with a brief overview of DOD-STD-2167A that describes the essentials of the standard as they pertain to HAC/RMPE. Then the tour proceeds to the "documentation tree," summarizing each document and providing a table of contents. In a separate presentation, the documents are linked to show requirements traceability from the topmost document, the System Operational Requirements Document, down to the source code prologues, illustrated with textual excerpts from each document (see figure 3). A similar tour shows the testability thread.

Hypertext Chapters

The hypertext chapters provide immediate access to software structure, functionality, and processing thread information. For software

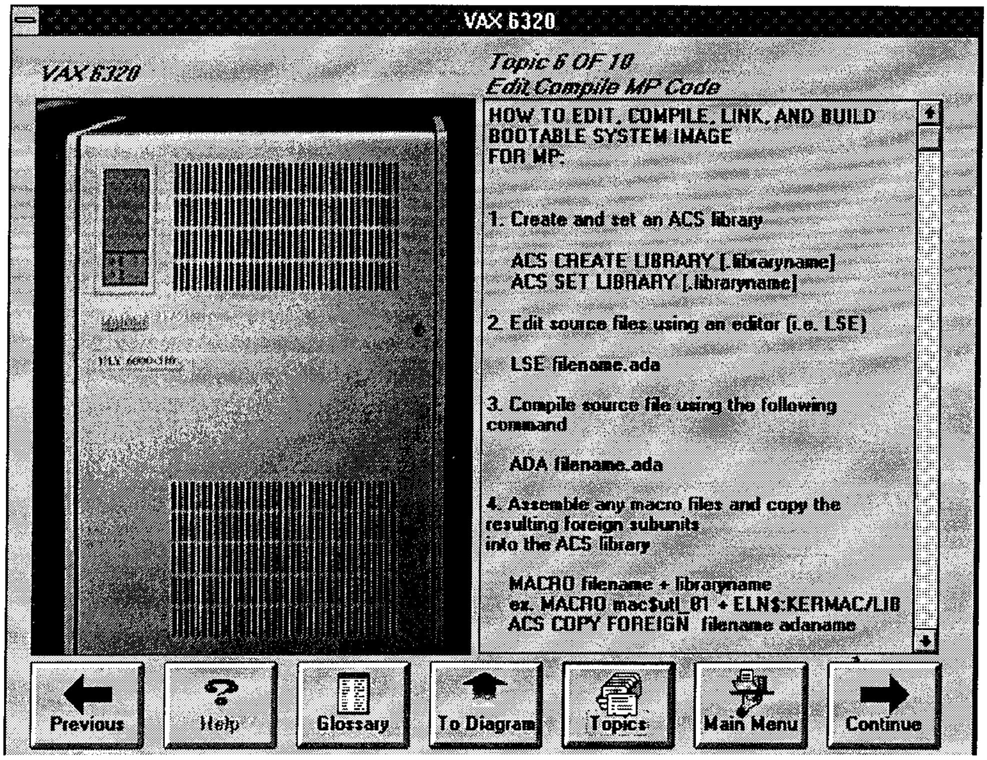


Figure 2. A screen from the Equipment Tour

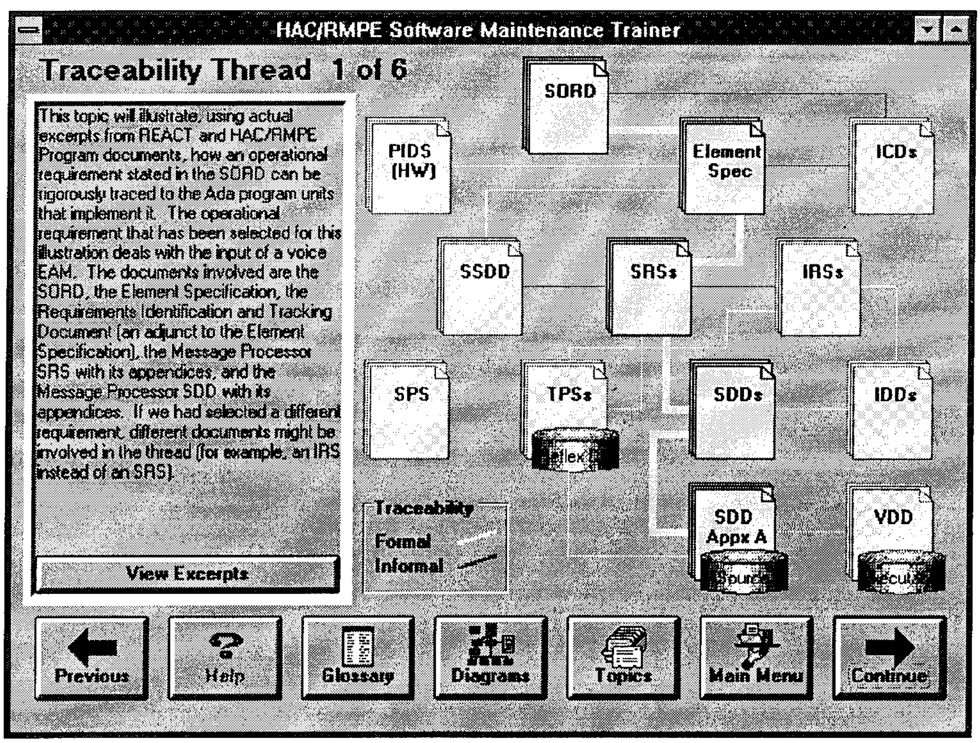


Figure 3. A screen from the Documentation Chapter

structure (i.e., the decomposition of the system into CSCIs, CSCs, and CSUs), a "MAP" button is provided that pops up a CSCI/CSC/CSU decomposition for reference. A detailed description of each hypertext chapter follows.

1. The *Human Machine Interface (HMI) CSC Chapter* enables the trainee to browse and explore HAC/RMPE's HMI design and implementation. The HMI for HAC/RMPE constitutes about 70% of the Message Processor software and is likely to see many modifications over the life-cycle of the HAC/RMPE software. We used a mockup of the HAC/RMPE operator console display as an index to the software components that implement it. For example, clicking on the Work Area of the console mockup brings up overview materials describing the types of operator tasks accomplished using the work area, the work area's "concept of operation." This information is followed by more information about the required functionality, and then the design of the work area's software components, including a decomposition of the work area software into its computer software components and units. Eventually, samples of the source code that implement work area functionality are available for browsing. The purpose of the HMI chapter is to help the new trainee build intuition about how the HMI software works by demonstrating how the HMI functions.

2. The *Data Storage Chapter* presents the software design of the Data Storage CSC. This software component includes the Adaptation Database, a section of code containing parameters that drive the screen displays, buffer attributes, and message processing algorithms. Since the intent of this design was to allow many of the software changes required of the HSSF to be accomplished by changing the values of data structures in the Adaptation Database, a chapter was created to train the software maintainer about the design of this system component.

3. The *Interactive Decode Chapter* includes a description of the Interactive Decode concept, along with history, design, and some implementation details of the functionality. This hypertext chapter includes screen displays to augment the functional description and tree diagrams to supplement the design description. A process thread is included to demonstrate the processing that occurs in decoding an emergency action message.

Practice Examples

The tutor prototype contains two practice examples that exemplify routine maintenance tasks that the HSSF will conduct. One example demonstrates a change to a message delimiter which correlates to a change in the Message Processor's Adaptation Database. Changing message delimiters and other related parameters will be required regularly. A second and more advanced example illustrates how a menu option is added to the HAC/RMPE HMI. All menus for the HAC/RMPE HMI are implemented using a common strategy, so by practicing adding a menu option the trainee gains insight for making other HMI menu changes as well.

These practice problems are presentation-oriented rather than interactive; they only require the user to watch the change taking place and read the explanations. The user views a sequence of steps, starting with a sample Software Change Request form describing the change to be made. The subsequent steps are appropriately illustrated with samples of source code and explanatory text, while a software change is made. *Before* and *After* samples of code are provided.

Navigation Aids and Context Sensitive Help

The tutor prototype contains on-line help and navigation aids to assure easy access to tutor materials. The Acronym and Glossary List provides quick access to terms and definitions.

The Help Guide describes how to use the tutor. The Diagram Tree serves as a navigation aid for the HAC/RMPE overview component.

Tutor Development Approach

The HAC/RMPE tutor prototype is designed as a collection of loosely coupled components, written in either Visual Basic™ or WinHelp™. Microsoft® Visual Basic™ is a visual language that facilitates the rapid development of Windows™ applications. Visual Basic™ provides a *Project* mechanism for encapsulating the functionality associated with each tutor component. Each Project is made into an executable file that runs independently in Windows™. Hypertext components are developed with the Windows Help System™ which comes bundled with the Microsoft® Windows™ operating system. The help compiler creates Help resource files that also run independently in Windows™. The MainMenu component, which serves as the tutor's on-line table of contents, is the framework that brings all the independent components together.

The prototype has an underlying general framework that can be applied to a variety of application domains. The MainMenu component, Overview component, and Tour components are easily instantiated for a different application domain other than HAC/RMPE. The front-end work required--gathering available information, extracting the useful and relevant information, and organizing the information into comprehensive training topics--is by far the most difficult part of building these components. The practice problems are designed more specifically to the problem domain.

The Microsoft® Windows Help System™ provides the necessary tools to build hypertext components. The process for building hypertext is fairly straightforward. Again, the most difficult part of constructing hypertext is the preliminary design work required. In addition to gathering information and organizing it into topics, hypertext design requires detailed analysis to determine how to link and navigate information.

CONCLUSIONS

The HAC/RMPE software maintenance tutor prototype, with its training system and on-line performance aid features, is a unique approach for supplementing the software maintenance training process. It is unique in two ways.

1. It captures the knowledge about the system mission, software structure and maintenance environment. This knowledge comes from a wide range of specialists working on the acquisition and development of HAC/RMPE, and is presented in a way that does not duplicate material already found in the deliverable documentation. Thus, it addresses the steep learning curve associated with a new maintenance trainee gaining proficiency with an unfamiliar project, software system, and maintenance environment.

2. It packages and presents information in a format, hypermedia, that to date is atypical for software maintenance. Hypermedia is an effective mechanism for dealing with complex relationships and multidimensional problems. Thus, with a subject such as software maintenance, where the job functions are not mechanical and cannot be precisely stated, hypermedia facilitates access to non-linear information and multiple contexts. Use of sound, video, and graphics motivates new staff members to learn, making the learning process stimulating and fun. Sitting down at the tutor and browsing information that is carefully composed into topics is less intimidating and more expedient than sitting down and surrounding oneself with large volumes of paper documentation, as is often the case with traditional systems and documentation.

In general, the concept of a system-specific software maintenance tutor as exemplified in the HAC/RMPE prototype shows great promise for filling a critical gap in the training of software maintainers and thereby reducing the life-cycle cost of system maintenance.

REFERENCES

1. Horowitz, Barry M., 1993, *Strategic Buying for the Future*, Washington, D.C.: Libey Publishing.

2. Kimmel, H. Steven, Dr., Office of the Under Secretary of Defense (Acquisition and Technology), August 1993, "The Plight of DOD Software Acquisition," Paper presented at the Test Technology Transfer Symposium, Burlington, MA.

3. "Defense System Software Development," DOD-STD-2167A, 29 February 1988.

INTELLIGENT EMBEDDED TRAINERS: A NEXT STEP FOR COMPUTER BASED TRAINING

Jonathan P. Gluckman, PhD
Intelligent Control Technologies Division of JJM Systems Inc.
Arlington, VA

Ruth P. Willis, PhD
Naval Air Warfare Center Training Systems Division
Orlando, FL

ABSTRACT

Acquiring the cognitive skills necessary to perform effectively as a member of a tactical decision-making team is neither a smooth nor a consistent endeavor. In order to extend training technology into a more dynamic domain we have created a system that utilizes expert defined problem solving skills and strategies, and compares them to those used by the trainee. Trainee models are inferred on the bases of monitored trainee behaviors and the use of probe techniques (such as verbal reports or questioning). Concurrence and divergence between the trainee and expert models, assessed as a function of outcome (was the answer correct and was it gained using a process similar to that of an expert), serves as the basis for feedback and skill building. Such systems could be embedded within the operational context to meet "train like you fight, fight like you train" requirements. This new generation of training systems is referred to as Intelligent Embedded Trainers (IET).

One ongoing program directed by the Naval Air Warfare Center Training Systems Division is to develop a standard, modular architecture for the development of IET systems. Critical aspects of the architecture include the use of a proven process model of human decision making and flexible knowledge engineering/artificial intelligence techniques in combination with structured training objectives, cognitive feedback techniques, performance assessment and tracking methods. The objectives of this paper are to describe the architecture used, outline the functional modes for development and operation of IET systems, and to demonstrate how the architecture addresses shipboard electronic warfare training.

ABOUT THE AUTHORS

Dr. Jonathan P. Gluckman is currently the Deputy Division Manager for the Intelligent Control Technologies (ICON) Division of JJM Systems Inc. located in Arlington, Virginia. He earned his MS in Experimental Psychology from the University of Cincinnati in 1988 followed by his Doctoral Degree in Human Factors/Experimental Psychology in 1990. Prior to his current position at ICON, Dr. Gluckman worked at the Naval Air Warfare Center Aircraft Division as a senior researcher and Block Program manager for basic and applied research in human factors.

Dr. Ruth P. Willis is a Research Psychologist in the Human Systems Integration Division of the Naval Air Warfare Center Training Systems Division. She received an MA in Industrial Psychology from East Carolina University and a PhD in Industrial/Organizational Psychology from the University of South Florida. Her research interests include individual and team training, and training technologies for distributed systems such as intelligent systems training.

INTELLIGENT EMBEDDED TRAINERS: A NEXT STEP FOR COMPUTER BASED TRAINING

Jonathan P. Gluckman, PhD
Intelligent Control Technologies Division of JJM Systems Inc.
Arlington, VA

Ruth P. Willis, PhD
Naval Air Warfare Center Training Systems Division
Orlando, FL

INTRODUCTION

Acquiring the cognitive skills necessary to perform effectively as a member of a tactical decision-making team is neither a smooth nor consistent endeavor, especially when the manner in which the US Navy conducts training is in transition. The Navy is rapidly moving to reduce its use of shore-based training facilities by increasing its employment of onboard training. To enable the next generation of sailors to acquire and maintain the cognitive skills they need, we must capitalize on advances in computer science and training technology for the Navy's deployed trainers.

Before we begin to look at what technology has to offer, we need to consider how learning takes place away from the schoolhouse. In most instances we learn by trial-and-error--usually participating in a great number of trials and committing numerous errors. How skillful we ultimately become depends in part on how well we can figure out what we are doing wrong and how to correct it. An alternate method is to find someone to serve as a mentor--someone with years of experience who can demonstrate the task, explain the steps along the way, watch us as we perform the task, and identify what we are doing wrong.

In order to assist the Navy in the delivery of high quality onboard training, how do we capitalize on the benefits of the informal training setting provided by a mentoring relationship and computer science? The approach we focused on was to combine intelligent simulation and training technology. Simulation technology was used to present the trainee with a realistic environment while intelligent systems technology provided the coaching and performance evaluation (cf., Lesgold, Eggan, Katz, & Rao, 1992).

Since current generation computer-based training (CBT) technology focuses on developing basic skills such as multiplication tables, electronic troubleshooting, and weapon capabilities, we were required to create our own training environment. In order to extend CBT technology into a more dynamic domain, specifically electronic warfare, we created a system that utilizes expert-defined problem solving skills and strategies and compares them to those used by the trainee. The trainee's understanding is inferred on the basis of monitored behaviors and the use of probe techniques. Convergence and divergence between the trainee's approach to solving the problem and the expert's approach serves as the basis for feedback and skill building. We refer to this training environment as Intelligent Embedded Training (IET).

One ongoing program directed by the Naval Air Warfare Center Training Systems Division is to develop a standard, modular architecture for the development of IET systems. Critical aspects of the architecture include the use of a proven process model of human decision making and flexible knowledge engineering/artificial intelligence techniques in combination with structured training objectives, cognitive feedback techniques, performance assessment, and tracking methods. The objectives of this paper are to describe the architecture used, outline the functional modes for development and operation of IET systems, and to demonstrate how the architecture addresses shipboard electronic warfare training.

THE KOALAS ARCHITECTURE

The heart of the IET concept is a process control architecture known as the Knowledgeable Observation Analysis-Linked Advisory System (KOALAS) (Barrett & Donnell, 1991). The KOALAS architecture provides support for human induction, incorporates an explicit model of the human operator's tactical situation assessment, and provides a context for the appropriate use of sensor fusion systems in the initialization and maintenance of that situation assessment.

In the KOALAS model, the sensor, decision formation, and action assignment processes are defined to be deductive in nature. The interpretation process, however, entails induction on the sensor data to generate the operative hypothesis for subsequent decision making and action. The most important issue in the design of human-mediated equipment control is the definition of the human operator's role in the sensing, interpretation, decision making, and action processes of the control system being designed. Since sensing, decision making, and action processes in the KOALAS taxonomy are defined to be deductive, these processes can be largely (or wholly) automated; it is in these areas that machine intelligence offers the greatest payoff in the control of multi-channel systems. The crucial human role in the system is in the interpretation process, a function that can be assisted, guided, or trained, but not automated (Willis, Becker, & Harris, 1992). This is the focus of training for IET systems.

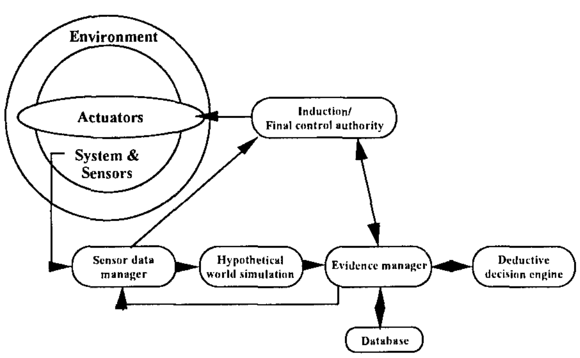


Figure 1. KOALAS process architecture

As illustrated in Figure 1, the KOALAS architecture incorporates an internal object-oriented simulation, called the Hypothetical World Simulation (HWS). In an IET application, the HWS is comprised of several major components: (1) a model of the world based on observed data (deductively generated and representative of ground truth); (2) a model of the subject matter expert's view of the world in terms of situation awareness, decision model, and actions (captured via artificial intelligence techniques); (3) rules for training opportunities such that cognitive skills can be exercised and appropriate measures of trainee behavior can be captured; (4) a model of the trainee including past performance and current competency levels. The function of the Evidence Manager and the Deductive Decision Engine in the architecture is to serve as the agents for comparison of trainee and expert models in order to provide feedback to the trainee, and to collect data on the trainee's responses and feed those back into the operating models in the HWS. Models which initially populate the HWS such as the subject matter expert, trainee history, and training objectives are stored in the Database along with specific information relating to the training application that the IET was developed to address (e.g. domain specific information such as radar signatures, platform and weapon types). The Induction box controls the dialogue and acquisition of information from the trainee. In the present demonstration, the HWS has been populated based on data from one subject matter expert. However, the KOALAS architecture can accommodate implementation of multiple subject matter expert models.

The IET demonstration that was produced was based on the model of human decision making presented in Figure 2. As can be seen in this figure, information in the environment impinges on a sensory system. Once processed by that sensor the information is sent for interpretation. The process of attaching meaning and developing a general model which accounts for the sensed data involves logical induction.

Several key distinctions reside in this model and were considered with regard to training in a dynamic environment. First, this model

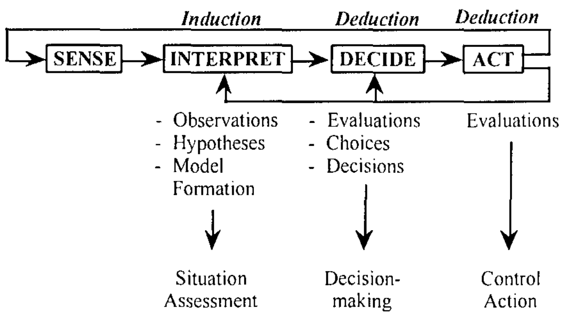


Figure 2. Human operator decision process model

presents a way of capturing where situation awareness occurs in the decision making process and it distinguishes between decision-making processes that are inductive versus those which are deductive. The degree to which a trainee masters the skill sets that allow efficient allocation of resources to those aspects of decision-making may be significant in distinguishing between novice and expert. It is toward that end that the development of IET systems are directed. To accomplish this it is necessary to incorporate the process model into the architecture of the IET. The IET is then built to incorporate techniques which can capture these aspects of subject matter experts and through the process of monitoring trainee behavior comparisons between trainee and subject matter experts are made and appropriate feedback is offered. Training efficacy is measured not solely in terms of "correctness of a decision" (e.g. correct product of a multiplication problem), but also in terms of the processes that the trainee uses to arrive at a decision. To accomplish this, IET utilizes a variety of artificial intelligence and other techniques.

THE EW APPLICATION

Electronic warfare is characterized by serious consequences, time critical identification, no clear right answer, deception, and incorrect and/or incomplete information. Effective task performance in this environment hinges on competent situation assessment requiring interpretation of sensor data to detect, classify, and identify threat systems and platforms, and to assess the threat's capabilities against ownship and/or other friendly forces. Yet, interpreting the sensor data is only half of the problem. Determining whether a particular airborne object is friend or foe may, under some conditions, depend solely upon the unknown pilot's intentions. And intentions, by their very nature, cannot be detected by sensors. Threat intentions, are, however, extremely important in the context of situation assessment and for selecting appropriate tactical action. It is to this task environment that we are looking to demonstrate intelligent embedded training.

The Electronic Warfare Intelligent Embedded Training (EWIET) environment is our proof-of-concept demonstration. EWIET runs on a 486 processor with an Orchid Pro Designer II graphics card and CD ROM and is written in C utilizing CLIPS for real-time artificial intelligence. EWIET emulated the graphical display of the SLQ-32 device, used by the EW community for ship self defense. The display of the device was fully simulated, so that the display changes as the trainee interacts with the device. How the trainee interfaces with the device however was modified due to the differences between a 486 PC keyboard and the SLQ-32's Fixed Action Buttons. As part of our demonstration we elected to incorporate a Navy-developed training scenario rather than develop our own. The scenario we used was designed by representatives of the Aegis Training community for use onboard Aegis equipped cruisers and destroyers. Within the context of this scenario, the trainees are required to identify the emitters which appear on their screen.

The KOALAS structure was embedded in the training system presented in Figure 3. The system functions such that prior to a training session, the system loads from the data base into the hypothetical model expert models comprising both deductive actions (in this case expert search patterns through different information sources) and inductive components (i.e., the general view of the tactical situation). This model once loaded runs in parallel to the training scenario being played such that it remains current. The system monitors the activities of the trainee, recording activities that are both consistent with and divergent from those expected based on the expert model.

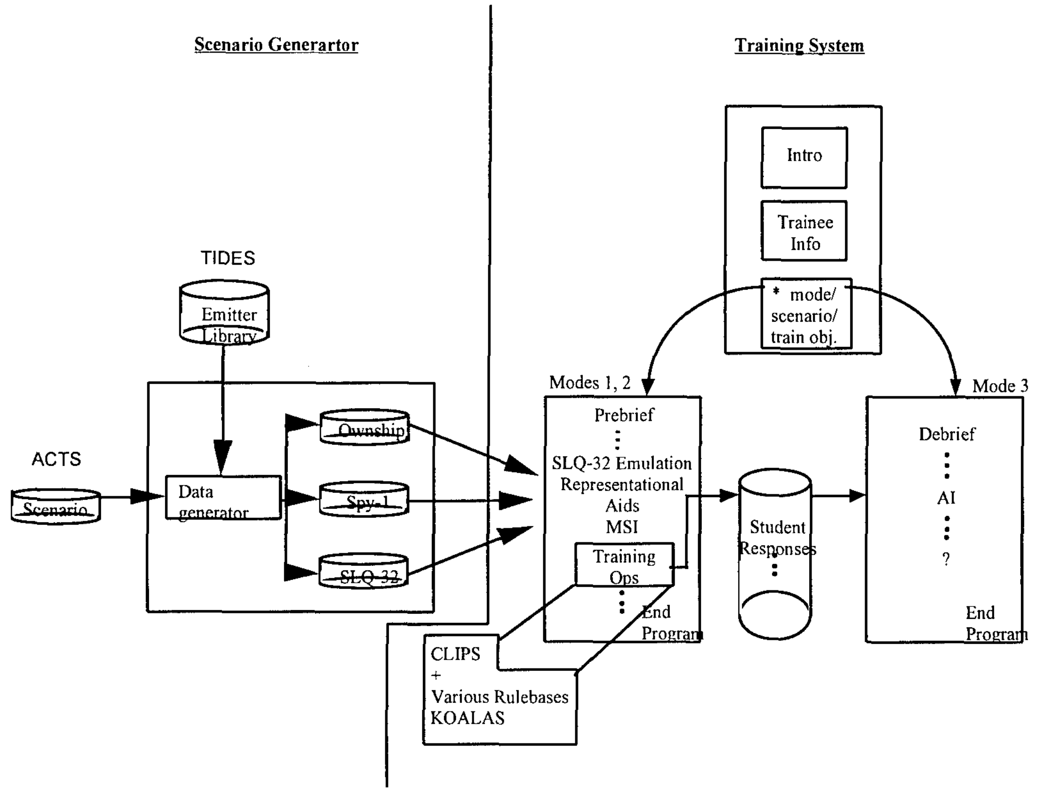


Figure 3. Electronic warfare operator training system design architecture overview

Three operating modes were developed to support acquisition, evaluation, and performance feedback. The first mode is a real-time free run mode. Using this mode uninterrupted trainee performance can be monitored and later evaluated and fed back to the trainee. The real-time mode affords an opportunity for the IET to be used non-intrusively within the context of ongoing operations. This capability is particularly important for embedded systems in which there is little time for the system to be dedicated solely to training and where the particular system is only one component of a larger group of systems.

While the real-time mode is useful for gathering information about deductive activities, inductive process are usually hidden from observable activity and can at best only be inferred from such activity. Thus, an interactive training mode (Mode 2) in which several techniques to isolate trainee performance and gather information specific to inductive processes, in this case situation assessment, was created. The interactive training mode utilizes probe techniques or interruption analysis to gather information related to inductive process. Using this technique, the training scenario is halted and the trainee is asked to perform activities or answer questions relative to specific training opportunities. Another novel aspect of the IET is the use of expert defined rule bases as part of the KOALAS hypothetical model used to evaluate the scenario in real-time for training opportunities. The specific rule base used during any given training session is determined as a function of the trainee's choice of predefined training objectives at set up time. This implementation provides for a very modular implementation which maximizes one's ability to utilize different artificial intelligence techniques to match training objectives and to conserve processing resources by only activating rules germane to the current training session. Moreover, this implementation also allows training to be tailored to individual needs and for all aspects of the training to be independent of the scenario used. From the stand point of the EWIET, this allows us the flexibility to use already established fleet training scenarios, incorporate new scenarios as they become available without changing the functionality or utility of the trainer, and to run in real-time (or live) exercises.

Once a rule set for detecting training opportunities is activated, the scenario stops to allow for training when the predefined conditions occur. At this point, the IET monitors observable trainee activities (keyboard inputs, etc.) until such time as the trainee keys that he has completed the task. Following correct emitter identification, a situation assessment battery patterned after the Situation Awareness Global Assessment Technique (Endsley, 1988) appears on the screen. These situation assessment questions specifically focus on gaining information about the trainee's current cognitive model and his inductive processes. Once the battery has been completed, cognitive feedback focused on presenting information to the trainee on the correctness of his response, what were the salient pieces of information for emitter identification and how to access that information, and what actions should/could have been taken. After feedback is given the trainee returns to the training scenario and proceeds until the next predefined training opportunity occurs.

Upon completion of the training scenario, trainees have access to a third system mode: Training Debrief. This mode focuses on giving the trainee a composite "report card" of his performance during the training session, direction for remedial activity, and information about how his performance differed from that of the expert. After the composite feedback, detailed feedback related to each of the training opportunities is available for trainee perusal. In order to compensate for potentially substantial time lags between when a trainee might complete the training session and when he views the debrief, facilities are provided for the trainee to view all critical displays and information as they appeared during the training session for each of the individual training opportunities.

Central to the use of such systems is the ability to clearly define training objectives. While this is not unique to training in general, it represents one of the key areas of stability from which an IET is developed. It guides the development of knowledge bases, the way knowledge engineering must be conducted, the choice of expert system used to train (rule based, CASE based reasoning, etc.), performance feedback, and accommodation of individual differences. The modular design of IET systems allows rapid incorporation of additional training objectives into the system as well as the associated additions to the artificial intelligence and feedback utilities. This flexibility provides for a variety of diverse training objectives to be accomplished within a single training system. Moreover, with many applications being performed on rapidly reconfigurable or generic computer workstations, it is likely that a single advanced IET system will be able to perform training on many jobs.

SUMMARY/CONCLUSIONS

This paper has described a process architecture known as KOALAS which has the potential to provide a unique environment for training complex cognitive skills. In a sense, where traditional CBT helps to develop "book smart" students, IET systems using KOALAS focus on transitioning "book smarts" into "street smarts." However significant challenges to the production of an IET exist. Most notable is the development of a standard or generic architecture which will the allow the system to fold around an existing operational system and be able to incorporate a variety of training techniques. The current concept demonstration has been useful in defining many of the basic functions and core aspects of IET systems but it has yet to undergo rigid testing and evaluation. The next step for the EWIET will be to evaluate the concept in an operational context.

REFERENCES

Barrett, C.L., & Donnell, M.L. (1991). Real-time expert advisory systems: Considerations and imperatives. Information and Decision Technologies, 16, 15-25.

Endsley, M.R. (May, 1988). Situation Awareness Global Assessment Technique (SAGAT). Proceedings of the National Aerospace and Electronics Conference (NAECON).

Lesgold, A., Eggan, G., Katz, S., & Rao, G. (1992). Possibilities for assessment using computer-based apprenticeship environments, p 49-80. In J.W. Regian and V.J. Shute (Eds.), Cognitive approaches to automated instruction. Hillsdale, N.J.: Lawrence Erlbaum.

Willis, R. P., Becker, D., & Harris, S. D. (November, 1992). Building a bridge between data fusion technology and training technology. Proceedings of the 14th Interservice/Industry Training Systems and Education Conference.

DETERMINING TRAINING RESOURCES AND REQUIREMENTS FOR NEW WEAPON SYSTEMS

First Lieutenant David M. Quick
Armstrong Laboratory, Human Resources Directorate
Brooks AFB, TX, USAF

ABSTRACT

This paper presents research being done to develop a training analysis tool that will allow training decisions to influence the design of weapon systems earlier in system development than ever before possible and to update these decisions throughout the system's life cycle. Integration of training into the acquisition and engineering process is often a very slow process. The Air Force has developed operational systems without qualified maintenance and support personnel assigned to the systems. Under current operations in the acquisition arena, funding is available for only a single training analysis. By implementing a method to influence design with training issues early in development, a trained and equipped force prepared to maintain and support new weapon systems will be available as the systems become operational. The objective of the tool is to select tasks for training, assign tasks to instructional settings, determine task training times, and determine training resource requirements for new systems by using an empirical data set associated with existing systems.

ABOUT THE AUTHOR

First Lieutenant David M. Quick is the Manpower Modeling Program Manager in the Manpower, Personnel, and Training Integration Branch of the Human Resources Directorate, Armstrong Laboratory, Brooks AFB, TX. 1st Lt. Quick received his B.S. in Operations Research from the United States Air Force Academy in 1991 and is currently working on his M.S. in Industrial Engineering/Operations Research at St. Mary's University in San Antonio.

DETERMINING TRAINING RESOURCES AND REQUIREMENTS FOR NEW WEAPON SYSTEMS

First Lieutenant David M. Quick
Armstrong Laboratory, Human Resources Directorate
Brooks AFB, TX, USAF

INTRODUCTION

Historically the Air Force has had difficulty performing timely analyses of the training resources and requirements for new weapon systems. The integration of training into the acquisition and engineering process of a new weapon system is a slow process and is often not initiated until design decisions have already been made. The determining factor for delaying training analysis is cost. Funding exists to perform the necessary training analysis only one time. Therefore, training analysis is shooting at a moving target because several, if not all, of the parameters influencing training change throughout the acquisition process. By delaying the training analysis, the opportunity to influence the design of the weapon system with training concerns and constraints is lost. In addition, the possibility of fielding an operational system without sufficient numbers of qualified maintenance and support personnel increases. Many tools already exist that can aid in the development of training, but the Training Resources and Requirements (TRR) tool stands alone as the only Air Force training tool imbedded in an integrated set of human systems integration analysis tools.[1]

NEED FOR AN INTEGRATED SET OF ANALYSIS TOOLS

Designing a new weapon system is a very complicated process with many tradeoffs taking place. The design process attempts to achieve the best blend of equipment, training resources and requirements, organizational and job structures, and individual worker-aiding technology domains to produce a weapon system able to meet its mission and performance requirements at the least cost. Figure 1 displays this complex process. If the training analysis is performed independently of these other domains, the value of the analysis diminishes. By including the training analysis in an integrated set of tools, training can be considered at the beginning of the acquisition process and can be used in subsequent tradeoff analyses to moderate training requirements of new systems. Integrated training analysis can automatically update training metrics as manpower parameters or personnel profiles change. Training parameters can be modified as the weapon system matures through the acquisition process to show training effects on system cost and performance.[2]

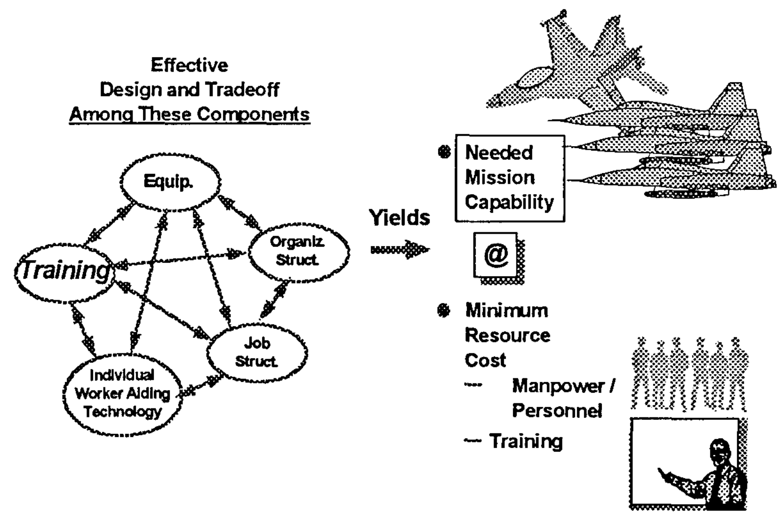


Figure 1 — Complex Process of System Design

MANPOWER, PERSONNEL, AND TRAINING IN ACQUISITION DECISION SUPPORT SYSTEM

The TRR tool is one of thirteen integrated tools in the Manpower, Personnel, and Training in Acquisition Decision Support System (MPT DSS). The MPT DSS is being developed to support Human Systems Integration during weapon system acquisitions by providing never-before-available analysis capabilities and reports, such as MPT cost estimates over the life cycle of a new weapon system. The MPT DSS allows the users (Program Managers, Integrated Product Teams, Operational Commands, and Defense Contractors) to examine the interaction of the various domains in the Air Force and to attempt to optimize system design by conducting tradeoffs between the domains.

Figure 2 presents the MPT DSS process. The first step of the process is to gather as much necessary data as possible so that it all resides on a single computer

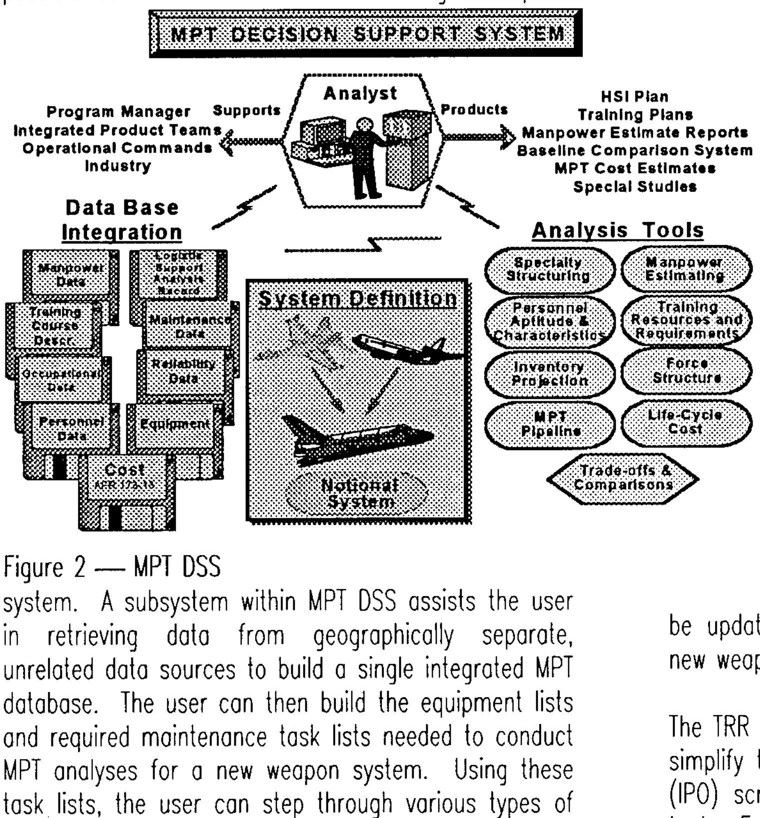


Figure 2 — MPT DSS

system. A subsystem within MPT DSS assists the user in retrieving data from geographically separate, unrelated data sources to build a single integrated MPT database. The user can then build the equipment lists and required maintenance task lists needed to conduct MPT analyses for a new weapon system. Using these task lists, the user can step through various types of analyses, such as a training analysis, using the analysis models within the MPT DSS. A combination of comparison and tradeoff tools allows for tradeoffs and sensitivity analyses to assess the impact of MPT and system alternatives.

Each tool in the MPT DSS relies on the other. For example, the TRR tool uses as input results from other tools to obtain data such as task difficulty ratings, projected inventory, and required man-hours to support the system by specialty. The data from the TRR, in turn, are used in analysis performed in other MPT DSS tools. The integration of the tools allows for the user to see the macro effects of making micro changes to individual models.

TRAINING RESOURCES AND REQUIREMENTS TOOL

Since the MPT DSS focuses primarily on the very early stages of the acquisition process, the TRR tool has narrowed its focus to Military Standard (MIL-STD) 1379D. The ultimate goal of this military standard is to enable the Government to identify more accurately the data or information the Government must have to fulfill a training requirement.[1] Since the standard has been prepared for joint service use, tailoring the model to this standard is critical to the acceptance of the TRR. The primary objective of the tool is to lay the foundation of the training required to maintain and support a new weapon system once it enters the operational inventory. To lay this foundation, the tool allows the user to select tasks for training, assign tasks to instructional settings, determine task training times, and determine training resource requirements. It has the capability to determine training requirements for all types of Air Force training (e.g., technical training, on-the-job training (OJT), and field training detachment (FTD)).[3] The tool is to be used primarily at the initial stages of acquisition, but it also has the potential and fidelity to be updated and used throughout the life cycle of a new weapon system.

The TRR tool exploits Microsof(TM Windows capabilities to simplify the user interface. The input/process/output (IPO) screen (Figure 3) is the roadmap for the TRR tool. Each ellipse or block represents a step in the analysis process. A color scheme identifies which steps have been accomplished and which is the most logical next step. The IPO also identifies the optional steps. An additional feature the tool provides is the "Notes" feature. Throughout the analysis, the user can document decisions made and why they were made. This leaves an audit trail to support the decisions when audited at a later date in the acquisition process. A new analyst would be able to pick up a study and use the audit trail and IPO to continue the study where it was left off. The TRR also provides an extensive "Help" system. The combination of all these aids provides the support necessary for a successful run through the model.[3]

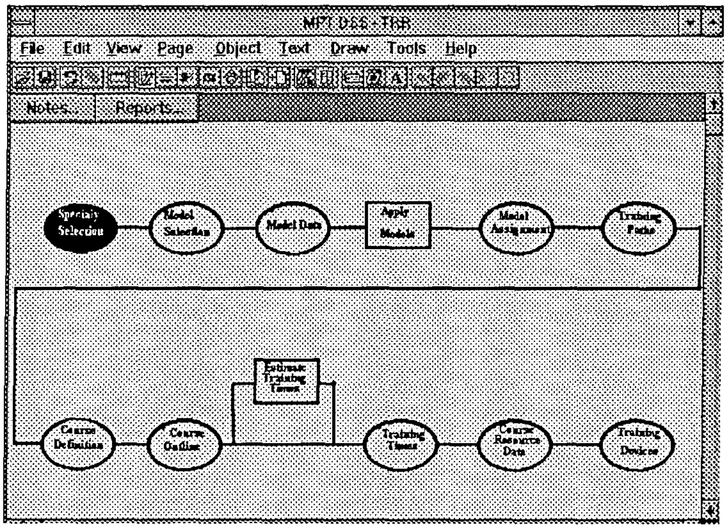


Figure 3 — TRR IPO

Inputs

As mentioned earlier, the TRR tool is just one of several tools in the MPT DSS. In fact, the TRR gets most of its data directly from the other tools. The Personnel Aptitude and Characteristic tool within the MPT DSS determines aptitudes and characteristics of the personnel. The Inventory Projection tool projects the personnel inventory for the life cycle of the weapon system and the Manpower Estimation tool simulates the required man-hours needed to maintain and support the system. The TRR uses all of this information in its analysis.[1]

Process

Specialty Selection - Before starting a training analysis, the user must determine which specialties to analyze. The TRR lists specialties and identifies which have courses already developed and the design differences categorized between the tasks performed by the new weapon system and existing systems. The user's concentrated effort should be the specialties with a highest degree of design differences. Courses required by existing systems will probably not change for those specialties that have a low design difference, so the required training resources will not change.

Task Selection Model - The TRR determines the tasks from the selected specialty that need training. The user will have a choice of four task selection models: 1) Training Emphasis, 2) Training Recommendation, 3) Multi-Factor Model, and 4) Train All. The Training Emphasis model bases its calculations on task factors from occupational surveys. The occupational surveys have a training emphasis factor, or a rating of which tasks require formal training for first-term personnel. The Training Recommendation model uses Logistics Support Analysis Record (LSAR) data. LSAR data have a single-position code indicating whether a task needs training and what type of training is needed. This training does not include equipment familiarization. The Multi-Factor Model allows the user the choose from a list of task selection factors and develop a new model. The following are the available factors:

- Training Emphasis
- Task Difficulty
- Percent Members Performing
- Average Percent Time Spent
- Mean Time to Repair
- Mean Operational Units Between Failure
- Hazardous Maintenance Procedure
- Task Criticality
- Training Recommendation.

The TRR obtains these factors from other tools within the MPT DSS or from LSAR data. The final task selection model is actually just an option to select and train all tasks performed by the specialty.[3]

Instructional Setting Selection - In addition to selecting tasks to train, the TRR must determine the setting to train these tasks. Each task can be trained by either formal training or OJT. The Air Force Occupational Measurement Squadron has created Automated Training Indicators (ATIs) for each task that help make training decisions using occupational survey data. The ATI value includes the percent members performing the task, training emphasis, and task difficulty. This value is used in accordance with the Course Training Decision Table in Air Training Command Regulation (ATCR) 52-22 to make 'level of training' and 'instructional setting' decisions. Air Force Utilization and Training Workshops have had success using the ATI value in making training decisions. Since the required data are already in the MPT DSS, it is appropriate to incorporate this proven model into the TRR. This model is an effective device to predict instructional settings for existing tasks.[4]

The TRR includes an additional instructional setting selection model for new tasks, the ATI Man-Hour model, since the percent members performing a task is not available for new tasks. It should be assumed that all members of a specialty will be performing the new task, so a better indicator is the predicted total number of hours spent per year on the task. The percent members performing is replaced with the annual hours spent on a task in the new model, but the ATI Man-Hour model still uses the other two indicators to stay consistent with the original ATI model.

The user interface for the instructional setting selection model allows the user to choose between the original ATI model, the ATI Man-Hour model, and manual selection. If the ATI Man-Hour model is chosen, the user must set a high and moderate ATI frequency cutoff value.

After selecting the task selection and instructional setting models, the user has the opportunity to view the data to ensure that appropriate data are available to support the analysis. If too many elements are missing, the user may want to reevaluate the chosen models. If the data look good, the user applies the selected models and receives a display of the results. The user will see a listing of tasks and the recommended instructional setting. At this point the user has the capability to override the outputs from the models. Overriding the suggestions from the models is a good reason to implement the 'Notes' function of the TRR. Anyone else who used the analysis in the future could call this function and read the rationale for overriding model results.

Training Pipelines - The next phase of the analysis is to develop courses and include them in a training path, or training pipeline.[3] The user can select an existing path to copy, edit, or delete, and can also create a new path. The user identifies each path by a user defined path name, start year, and end year. If the new path is very similar to an existing path, the user should just copy the existing path. By copying a path, all the courses and course information is available for modification. The rest of the work within the TRR uses the selected path until the user returns back to this point and selects another path.

After determining the training pipeline, the analyst defines the courses. The user assigns courses to the training path. Three different types of courses can be added to the path: technical training, OJT, and other (e.g., FTD, Mobile Training Teams, Professional Military Education, Career Development Course, and correspondence). The user can use the New System Training Plan or review of existing courses (AFM 50-5, Training Management System) to identify courses.[1] The user can copy courses from a list of established courses, and then make modifications. This prevents duplication of work that has already been performed.

The user must define a list of elements for each course. Depending upon the type of course, the tool needs different combinations of these elements. The first element is the course number (Figure 4) which includes the responsible training center, training type designation, students for which designed, planned area

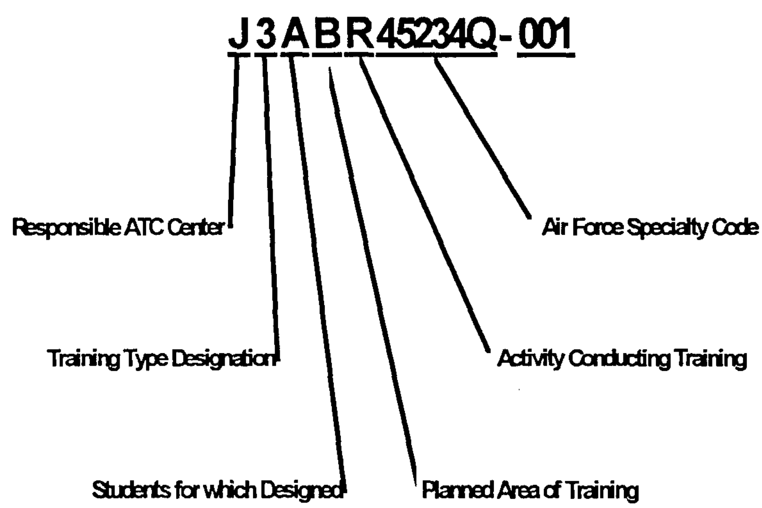


Figure 4 — Sample Course Number

of training, activity conducting training, and Air Force Specialty Code (AFSC). The course numbering system used is the standardized method of numbering courses, according to Air Training Command Regulation (ATCR) 52-21. The TRR library system lists this information to aid in the development of course numbers. The other elements that make up the course definition include the course title, skill level, security class code (in the TRR libraries), preceding course within skill level, and whether or not the course is weapon system specific. The user can also determine whether or not to calculate resources from any given course. The TRR only generates automated training cost summaries for resident special training, resident regular training, and field technical training.[3]

When defining the courses, the user assigns each task requiring training to a course. The TRR provides a list of tasks selected for training for the given specialty with their appropriate courses. The user assigns each task to a single course. The user will have to define enough courses to train all tasks requiring training.

With the course definitions complete, the TRR builds the training pipeline (Figure 5) for the given specialty by using the course skill level and preceding courses. The training pipeline is a graphical representation of

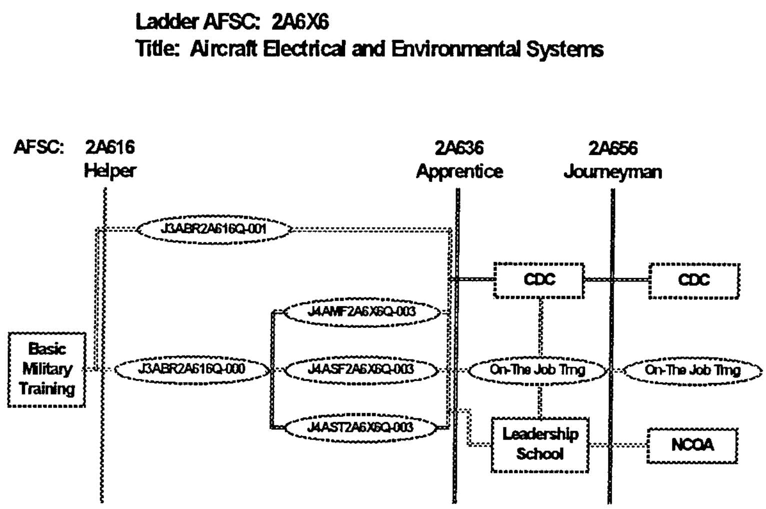


Figure 5 — Training Pipeline

the training requirements for personnel within a given specialty. By viewing the pipeline, it is evident what courses a career path require and at what skill level they are required. The TRR uses the training pipeline interface to access the different courses for the remaining steps of the process.[4]

Course Outline – After creating courses and assigning them tasks, the user must go back and begin to fill in the details by developing the course outlines. The goal of the TRR is not to build course curriculums and objectives but to determine the required training resources and requirements for cost analysis and tradeoff studies.[1] For example, tasks can be realigned within the specialties of a new weapon system, and the analyst may want to know the implications of making such changes. An analyst must be able to determine resources required for each course when these tasks are reshuffled. The course outlines only have to be at enough detail to determine the required resources to train the tasks.

For system specific courses, the courses break down into blocks and units. The tool can assign tasks to these course modules, which allows for analysis in greater detail for those courses that are high drivers in the new weapon system. For non-system specific courses, the TRR estimates resources without the in-depth analysis. Other weapon systems influence their course outlines.[4]

Training Times – Now that tasks are assigned to the courses, blocks, and units, the TRR helps determine training times and methods/media for the tasks. The TRR tool allows for the user to enter the task training times manually, but it also has the capability of estimating the required training times. It uses modified functional relationships from the Training Analysis Support Computer System (TASCS), developed by the Training System Program office at the Air Force's Aeronautical Systems Center, to predict required training time.[5] The TRR applies each task to the various TASCS task learning categories (Figure 6) in order to calculate the task training time. The TASCS equations use information that the TRR and MPT DSS already provide for other calculations. The training times coming out of these equations are adjusted to reflect the skill, knowledge, and ability similarities between tasks within a unit, block, or course, depending upon the lowest level course module with assigned tasks. The TRR uses a table containing the percentage of time used for each method and media by course type to divide the total training time into method and media assignment. These tables are accessible and modifiable by using the TRR libraries.[4]

After all of these calculations are complete, the user has a chance to review the results. The results are only recommendations and the user has to input the final training time. The user may be a training expert and choose to override all calculated times, but these calculations are available to assist in making educated estimates.

Task Learning Category	Individual Methods/ Media of Instruction	Calculations for Task Training Times
Explanation	Classroom Lecture Questioning Discussion Self-Directed Audiovisual Media	Task Difficulty x 60 min.
Demonstration	Classroom Demonstration Demonstrator	Task Difficulty x Task Length x 1.5
Cognitive Part Task	Laboratory Programmed Questioning Procedure Trainer Panel Trainer Interactive Courseware	Task Difficulty x Task Length x Task Criticality x .2
Psychomotor Part Task	Laboratory Hands-on Performance Practical Exercise Part Task Trainer Operational Equipment	Task Difficulty x Task Length x Task Criticality
Full Task	Laboratory Operational Environment Training Exercise Simulator	$(\text{Task Difficulty})^2$ x Task Length + 1

Figure 6 — Task Training Time Functions

Course Resource Data – To assist the user in defining the course resource data, the TRR provides a library of historical courses from Air Education and Training Command's Programmed Technical Training data base allowing the user to find a comparable course. The required course resource elements are:

- Course Length
- Attrition Rate
- Programmed Group Size
- Academic Weeks Per Year
- Percent Permanent Change of Station (PCS).

Once again, the user can use data from an existing comparable course, or the user can input new data. Each of these elements influences the training costs of new systems.[4]

Training Devices – The final step of the TRR process is to define the training devices necessary for the training media. In an earlier calculation, the training times were determined by method and media. In order to put a cost on the training media, the TRR assigns training devices to each media. For each training device, the user enters its developmental cost, cost per student hour, and the hours spent using it per course.[4] Once these data are entered, the process is complete for the course being analyzed. At this point, the user can build another course for the current training path, build a new training path, select a new specialty to analyze, or end the TRR analysis.

Libraries

Throughout the discussion of the TRR process, several references were made to the TRR libraries. The libraries store the information that will be consistent across all TRR analyses. This information includes:

- Training Methods
- Task Selection Models
- Media Types
- TASCS Categories
- Course Location
- Course Types
- Student Types
- Training Activities
- Training Areas
- Course Security Classification
- ATI Library
- Media Classes.

All the libraries, except the ATI Library and Media Classes, are editable by the user. Since these libraries are constant across all TRR analyses, the edits made by the user are saved as a permanent part of the library.[3]

Reports

Another feature available in the TRR is the report function. Figure 7 shows the available reports. These reports display all of the information that the user may need to give to others in a more presentable format.

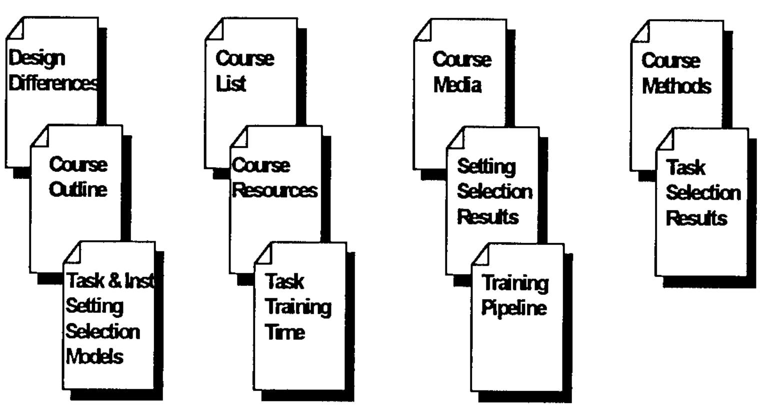


Figure 7 — TRR Reports

Outputs

Since the MPT DSS is an integrated set of tools, other tools will use output from the TRR. More specifically, the Force Structure tool that calculates manpower and develops the force structure required to support a new weapon system. The TRR feeds this tool the course information (i.e., course length, number of students, course setting, and method of training) so it can calculate the instructor requirements.

The goal of the Life Cycle Cost tool in the MPT DSS is to estimate the cost of MPT elements over the life cycle of a new system. It pulls together all the resource information to develop estimates of the MPT life cycle cost. The TRR feeds this tool with the complete course information, including training devices. With this information, the MPT DSS can estimate the training costs for the life cycle of new weapon systems, as well as the complete MPT costs.

PAYOFFS

The potential payoffs of including a tool such as the TRR in an integrated environment are great. Any design, personnel, or policy decisions made on a new weapon system have an effect on training, and this effect can now be reflected without starting a new analysis for each option. For some changes, the analyst will have to refine the TRR analysis, but for others it will update automatically. This allows an analyst to perform a training analysis during the early stages of the acquisition process so that training is included in the tradeoff decisions of designing a new weapon system.[2]

Another advantage of being in an integrated set of tools is that simple changes can be made within the tool, such as changing on-site classes to distant learning, and the cost implications are easily calculated. This allows for training optimization tradeoffs to be performed after the training analysis is complete.

SUMMARY

The TRR is a tool that provides the capability to conduct training analysis earlier in the acquisition process than ever before. By using the structured analysis approach of the TRR, an analyst can develop training paths for the different specialties required to support a new weapon system. The training paths will include course outlines with training times for the different methods and media for training. The MPT DSS pulls together these requirements and resources in the Life Cycle Cost tool to show the influence of training on the life cycle costs of the new system. Not only will this allow for a capable force to be trained and equipped when the new weapon system becomes operational, but an analyst can perform analysis early enough to have influence on the design and policy decisions throughout the acquisition process.

REFERENCES

[1]Dynamics Research Corporation (1992). Management Plan for the Development of a Prototype Manpower, Personnel, and Training (MPT) Decision Support System(DSS). Brooks AFB, TX: Manpower and Personnel Division, Human Resources Directorate, Armstrong Laboratory.

[2]Dahn, D.A., Sorensen, H.B. (1993). Manpower, Personnel, and Training Analysis in Aerospace System Development. Proceedings of the 15th Interservice/ Industry Training Systems and Education Conference (I/ITSEC 1993). pp 10-23.

[3]Dynamics Research Corporation (1993). MPT DSS First Annual Review. Handouts. Brooks AFB, TX: Manpower and Personnel Division, Human Resources Directorate, Armstrong Laboratory.

[4]Dynamics Research Corporation (1994). MPT DSS Second Annual Review. Handouts. Brooks AFB, TX: Manpower and Personnel Division, Human Resources Directorate, Armstrong Laboratory.

[5]Logicon, Inc. (1987). Training Analysis Support Computer System (TASCS) User's Guide. Wright-Patterson AFB, OH: Aeronautical Systems Division, Air Force Systems Command.

THE FUTURE OF SELECTIVE FIDELITY IN TRAINING DEVICES

Dee H. Andrews, Lynn A. Carroll, Herbert H. Bell
Aircrew Training Research Division, USAF Armstrong Laboratory
Williams Gateway Airport, Mesa, Arizona

ABSTRACT

Since the inception of modern simulation the designers and users of training devices have attempted to replicate as many physical and functional stimuli as possible in the training device. There are three primary impediments to this activity: our frequent inability to specify the kinds of stimuli that are required, our technological difficulty in replicating some stimuli, and the cost of replicating stimuli.

The constraints cited above have led the training device community to develop the concept of "selective fidelity", meaning that we have to be very selective about the stimuli that we choose to replicate. This paper presents arguments that our definitions of selective fidelity now need to be altered to fit recent behavioral and engineering developments. Over the years we have improved our ability through research and analysis to define the important stimuli. Also, our engineering capability to replicate formerly difficult stimuli has improved significantly. Finally, there have been dramatic decreases in the cost of providing high fidelity simulation. In this paper we discuss our belief that while the concept of selective fidelity will remain important to the training device community, the definition of selective fidelity will be more focused on trainee learning requirements than on analytical and technological shortcomings.

Biographies

Dr. Dee Howard Andrews is currently the Technical Director of the Aircrew Training Research Division, Air Force Armstrong Laboratory. He previously was a research psychologist for the Army Research Institute and the Naval Training Systems Center. He received his Ph.D. in Instructional Systems from Florida State University. His research interests include; instructional systems development, training device design and use, training system evaluation, and training cost-benefit analysis.

Colonel Lynn A. Carroll is a Command Pilot with extensive experience in tactical fighters and training aircraft (AT-33s, A-7Ds, T-39s, AT-38s, and F-15s). He is currently the Division Chief of the Aircrew Training Research Division, Armstrong Laboratory. Colonel Carroll graduated from the University of Iowa with a BA/MS and was commissioned through the AFROTC program.

Dr. Herbert H. Bell is a Research Psychologist with the Air Force Armstrong Laboratory. He received his Ph.D. in Experimental Psychology from Vanderbilt University in 1974. Dr. Bell's current research focuses on the use of distributed interactive simulation for combat mission training.

THE FUTURE OF SELECTIVE FIDELITY IN TRAINING DEVICES

Dee H. Andrews, Lynn A. Carroll, Herbert H. Bell
Aircrew Training Research Division, USAF Armstrong Laboratory
Williams Gateway Airport, Mesa, Arizona

INTRODUCTION

Since the days of the early Link trainers the simulation community has striven to re-create for trainees a virtual world that is as close to the real world as possible. To do this, training device designers have attempted to replicate the stimuli and response modes that trainees would eventually see and use in the real world. Despite these efforts training devices always have fallen short of replicating a complete set of the outside world stimuli.

A variety of problems have plagued these efforts. Designers and instructors have frequently had great difficulty in defining precisely the characteristics of the stimuli to be presented to the trainees. When training requirement experts have been able to adequately define the stimuli necessary for a task, the designers have frequently been unable to replicate the stimuli because they lacked the technology. Sometimes, the technology has existed, but the procurement and/or sustainment costs have been too high.

Selective Fidelity

The end result of the problems cited above has been an approach referred to as "selective fidelity". That is, since we can't simulate all of the stimuli that are present in a real-world task or function, we must then choose to simulate only those stimuli that truly are necessary to perform the task. We <u>select</u> the <u>fidelity</u> level that we will present to the trainee. While this approach has proven beneficial, especially in developing tactical training devices, the problems cited earlier can greatly hamper this approach.

There are two main types of fidelity: physical and functional. (Other types of fidelity are sometimes discussed, such as task, behavioral, psychological, equipment. However, for purposes of this paper we will focus on the two types most commonly discussed: Physical and Functional.) For a complete discussion about fidelity types we recommend (1).) Physical fidelity refers to the physical lay-out of the operator's simulated environment. Are the knobs, dials, displays, controls and so forth present and in the correct positions? Are the visual stimuli present in the kind and degree that we would expect to see in the real world? In other words, does the synthetic environment "look right"?

Functional fidelity refers to the way the simulation acts and its similarity to the real world. For example, does the simulator respond to the operator's control inputs in the way the actual equipment would? Are the aerodynamic or hydrodynamic equations accurate? In other words, does the simulation "act and feel right"? Functional fidelity is very much focused on the stimulus presented to the operator and the response that the operator makes.

For every simulation task we have an array of fidelity levels to choose from, but we typically are not able to reach the real-world level of fidelity. Our inability to precisely define stimuli and technology/cost limitations conspire against us. We must then select the fidelity level that will allow us to meet our simulation requirement while still conforming to our technology and budget constraints.

This selection process requires a number of trade-offs between the training requirement, the technology available, and budget. The fact is we seldom are able to select the fidelity level that we truly desire. We usually must settle for the level that will allow the training mission to be accomplished affordably.

In training operators in psychomotor tasks we generally feel that the quality of the training will increase with each incremental increase in physical and functional training fidelity. In operator training we strive to "select" a fidelity level that will replicate as many of the physical requirements and functional fidelity as possible, including trainee response modes. Research and experience have shown that relatively lower levels of fidelity are required for familiarization and cognitive training. Tactical training is an example of cognitive training.

Examples of Selective Fidelity

Selective fidelity has proven most popular as a concept in the development of tactical training devices such as the Advanced Research Projects Agency's SIMNET (2). The goal in SIMNET was to develop a networked set of simulators that would allow tactical commanders at the battalion level and below to learn how to tactically deploy and fight their armored units. Only those displays and controls deemed absolutely necessary for tactical training were provided to the tank commander, gunner and driver.

Cost was a major factor in deciding not to replicate all of the controls/displays and functions from the real world armor. Since tactical training of the maneuver element was the objective, SIMNET designers decided it was better to spend money on many SIMNET devices with lower physical and functional fidelity than it was to spend money on fewer devices with higher fidelity. Since each physical feature and function that is replicated costs additional money, the only way to provide enough devices to satisfy the objective was to select only those features and functions that were necessary for the tactical tasks.

The result was the development of dozens of SIMNET devices that allow an entire battalion to train together. If the selective fidelity approach had not been used, and higher levels of fidelity had been selected, the Army and ARPA would have only been able to provide a relatively few devices. The result would not have provided the trainees enough manned armored units to achieve the objective. At the time of SIMNET development the designers referred to their approach as the "60% solution". Meaning that only about 60% of the physical fidelity was provided to the trainees that they would find in their actual armored systems. Experience has shown that for tactical armored system training approximately 60% of the physical fidelity seems to be adequate. Functional fidelity was higher than 60% for SIMNET, but still something less than 100%.

The Army provides much higher fidelity trainers for training gunnery skills to gunners and tank commanders in the Conduct of Fire Trainers (COFT). The COFT devices cost considerably more per unit than do the SIMNET devices, but they provide a higher number of the physical and functional stimuli a gunner and commander would expect to see in the real world. In addition, there are many more input controls than are provided in the SIMNET devices. This level of fidelity is required for training gunners to hit their targets and for training commanders and gunners to coordinate. It is assumed that basic gunnery co-ordination skills are already possessed by commanders and gunners when they start to learn broader tactical skills in SIMNET. If SIMNET focused on the 60% physical fidelity solution then it may be fair to say that COFT focused on the 80%+ physical and functional fidelity solution for gunnery tasks. The physical and functional stimuli for gunnery training provided in COFT may not exactly match the real world stimuli, but they come fairly close. The functional fidelity needed for tactical armored maneuver element tasks was much lower in COFT than in SIMNET because tactical maneuver element tasks were not intended to be trained on COFT.

It should be pointed out that even in COFT a selective fidelity approach was used. Detailed training device front-end analysis showed that not all of the real world stimuli were required to teach the gunner and commander their tasks. Therefore there was no attempt to include those stimuli. In addition, there were some stimuli that were deemed important, (e.g., the ability for the commander to look outside the open hatch) but they were not replicated for one of three reasons: 1) they cost too much to simulate, 2) it was not technologically feasible to simulate them 3) it was impossible to define all the possible stimuli due to limitations in our behavioral analytical techniques. Such occurrences happen in <u>every</u> training device development.

For some time now the success of the 60% solution SIMNET approach has been touted as the optimal way to proceed for all simulated tactical training tasks. The argument has been that we simply either can't afford to reach for much higher levels of fidelity in tactical training because it is too expensive or it is not necessary. This argument says that we can achieve quality tactical training by making use of the selective fidelity approach. As the DoD moves more and more into using networked trainers for service unique and multi-service training, there is continual interest in replicating only those stimuli and control inputs that are deemed absolutely essential for tactical training.

There should be some concern with absolute acceptance of the 60% solution approach as the optimal method for achieving tactical training in networked environments. We have found in tactical aircraft training that a considerably higher level of fidelity is required than 60% for some pilot training functions. Single-seat fighter pilots must perform; the command function that a tank commander has to perform, the vehicle control function that the tank driver has to perform, and the weapons deployment function that the tank gunner has to perform. While the command function may be able to use the 60% physical fidelity level, we have discovered that the weapons control and deployment functions need higher physical and functional fidelity levels.

Aircrews developed their own response to achieving desired fidelity. Frustrated by the laborious and lengthy process of analysis and media selection, the user simply stated, "make it look and act like the weapon system." The solution, although expensive, was to literally "cut off the front third of an aircraft" and stimulate it. However, while high fidelity cockpits were achievable in this way, the complementary synthetic environment that would permit a full range of training was not so easily achieved. Simulators supporting single-ship missions evolved to accommodate a full spectrum of training from individual task to crew mission training. Simulators supporting fighter missions also evolved to achieve full cockpit fidelity, but typically were designed to support individual, procedural training at the task level. While excursions were made using full motion and visuals, fidelity limits did not permit credible multiship mission training. In both applications, families of trainers were used to augment training at reduced fidelity levels and cost.

However, it was the fuel crises of the 1970s that precipitated the push for simulation to actually replace flying hours that created the real dilemma. How much fidelity was enough? How many simulator hours were required to replace an hour of flight time? While these questions have yet to be conclusively answered analytically, they are being overcome by events. Mission training requirements are escalating while flying hour erosion continues. Classic peacetime constraints such as funding, safety and security are now complicated by environmental constraints and training area encroachment. The results are an expanded use of simulation and acceptance by many aircrews of simulation as a potential solution. However, this should not be construed to mean that simulation fidelity is now adequate or that selective fidelity is the total answer.

The same factors that forced designers to make selective fidelity decisions (cost, technological problems, and difficulty in defining all possible stimuli) still plague fighter simulator designers. Selective fidelity is still required, but the ultimate level of fidelity will vary depending upon the domain of training.

FUTURE OF SELECTIVE FIDELITY

We believe that the concept of selective fidelity will always be a major concern of the training device community. However, we also believe that our understanding and use of selective fidelity will need to change as a result of a number of current trends. Following is a discussion of three trends.

Behavioral and Cognitive Research Contributions

For many years the primary method of defining which stimuli to simulate was primarily an analytical one. That is, analysts would observe experts and journeymen performing their jobs and attempt to determine which stimuli in the environment were prompting certain types of responses. In addition, the analysts would talk with those same experts and journeymen to get their opinions of what was required to perform the task. There was some basic research about human performance that helped, but the analytical technique was still the prime method (3).

This reliance on primarily analytical data rather than experimental data has been a major reason for adapting a selective fidelity approach to training device design. In many cases the training analysts and psychologists were simply not able to specify a complete set of stimuli for successful task performance because they didn't have the tools. This difficulty has been a major impetus for using a selective fidelity approach. In fact, in many cases it has proven difficult even to specify all of those stimuli which are crucial.

The empirical research that was performed concerning vital stimuli was directed at overt stimuli and response characteristics of the tasks. Researchers would provide various sets of simulated cues and then attempt to measure trainee responses to see how closely they matched responses from the real world. If the responses in the experimental conditions corresponded relatively well to the responses in the real world, it was assumed that the simulated stimuli were then also faithful representations. What made the experimental stimulus a faithful representation was not often discovered. A major reason for this lack of discovery was that the psychological techniques, theories and measures necessary to make those discoveries did not exist or were not adequate.

In recent years there has been new thinking about how humans learn and perform. Cognitive approaches to examining learning and training requirements have begun to be common in the training analysis business. These cognitive approaches, such as the Integrated

Task Analysis Methodology (4) and others in references 5-7 as examples, allow analysts to look at more than just the external stimulus and response conditions of a real world task. In addition, they can provide an insight into the actual thinking processes that an expert or journeyman use in performing their real-world tasks. Anyone who has performed training system analysis with experts and journeymen realizes the difficulty often inherent in "extracting" verbal descriptions of the stimuli that are required for successful task performance. Traditional Stimulus-Response (S-R) type analyses are only partially helpful in this articulation process. New cognitive techniques build upon existing analytical techniques to provide a better "window" into the expert's and journeyman's discrimination and generalization of stimuli.

Cognitive techniques also provide a better understanding of the cue recognition/selection process. Stimuli to the five senses surround us every moment that we are alive. We have a marvelous capacity to filter out from our conscious attention the vast majority of these stimuli. Our information processing system performs this filtering by analyzing each stimulus and comparing it to a huge store of memories from our long term memory. A pattern recognition process then allows us to make a decision about whether a stimulus has meaning for us. In other words, is the stimulus to be treated as a cue for the expert or journeyman to make some type of response?

By better understanding the cue recognition/selection process, training analysts can now more easily define the complete set of stimuli that must be replicated in the training device. In the past, analysts had to make use of selective fidelity because the classical S-R theories and tools necessarily limited a definition of the complete set. Selective fidelity was almost forced upon the training device definition process because of the difficulty of defining anything more. Cognitive analysis shows great promise for defining the relevant stimuli for various tasks. This will improve our ability to specify physical and functional fidelity levels.

Improved Technological Expertise

The early Link flight trainers represented only a very few of the real world stimuli that were available to a pilot while flying the aircraft. That was largely because the engineers of the time simply did not have the technology available to replicate many of the stimuli. Through the years we have greatly added to the repertoire of stimuli which can be simulated. Some examples are:

- digital audio systems now allow a fairly accurate representation of the weapon system, communication and battle sounds that a warrior can expect to hear in the real world.
- visual systems (image generators, displays, data bases) have greatly increased the training designers capability to present realistic visual stimuli.
- control loading technology has enabled trainees to experience realistic tactile and proprioceptive feedback.
- considerable progress has been made in developing motion systems that fairly represent motion cueing. (However, motion system simulation is a good example of relevant cues and technology limits for tactical aircraft training. Research and experience failed to show the effectiveness of motion systems for training tactical fixed-wing aircraft tasks. Analysis showed limited ability to replicate the range of motion and sustained motion. Selective fidelity with force cueing, such as G-suits and G-seats, were just as effective for on-set motion cues.)

Despite the technological progress that has been made, the training device community realizes that there is still a considerable distance to cover before we can say that we are approaching 100% fidelity in the areas described above. We believe that the next ten years will herald a rate of engineering breakthroughs that will easily match or exceed the rate of improvements seen since the Link days. A very high degree of replication of real world stimuli will be the rule and not the exception, as is the case today. Image generators, visual displays, data bases, networking, motion bases, control loaders and a variety of other simulation tools will all make dramatic advances. These technological tools, and methods will allow training device designers to worry more about identifying a complete array of stimuli necessary for successful training objective accomplishment, and less about which stimuli are absolutely essential.

Microprocessor Cost Reductions

Costs of producing higher levels of fidelity have declined dramatically in the last decade. These price declines are due to the advances in microprocessor technology that have affected large sectors of society. The cost to produce various simulated stimuli (visual, aural and proprioceptive) is obviously driven by the amount of computing power required.

Perhaps the best way to illustrate the effect of computer cost reductions on selective fidelity is by using an example from the tactical training domain. Compared to the visual imagery available at the time SIMNET was developed, the SIMNET imagery had a fairly impoverished visual scene content and resolution. The design logic was that for tactical training it was not necessary to replicate large numbers of visual stimuli or provide detailed resolution. It was enough for the tactical decision maker to know that a friend or foe was in a certain place. It was not of vital importance that high resolution be provided in order for tactical training objectives to be met.

In the future however, image generators (IGs) that can provide extremely high levels of visual fidelity will be available at affordable prices effectively reducing dependence on selective fidelity. The price of a channel of simulated, high fidelity imagery has been dropping by about 75% every five years over the last ten years. We would expect that cost reduction trend to continue, and in fact accelerate, over the next ten years. Each tactical training device will have present the same degree of resolution to tactical trainees that has here-to-fore been available only to operator trainees (e.g., the tank gunners in a Conduct of Fire Trainer). The selective fidelity decisions that are made will not be driven by cost, but rather by factors more related to the actual training objectives, such as the phase of training and our ability to define real-world stimuli.

One result of this greater availability of high resolution visual imagery will be the demise of separate trainers for operator vs. tactical training. One simulator will be able to accomplish both of these goals. Tremendous cost savings will accrue from being able to train both sets of skills in the same "box". Large savings will be based upon reduced unit production costs because more of the same boxes will exist. Where before the SIMNET designers had to pick only those stimuli to simulate which were absolutely crucial to the tactical training task, now designers will be able to select from a much broader range of potential stimuli because the cost of simulation will be far less.

CONCLUSION

The selective fidelity concept has served the training device community well. It has allowed us to cope with analytical, technological, and cost constraints and yet still produce many training devices that were effective. Unfortunately, there were also many training devices that did not allow their training audience to reach the training goal completely.

We believe that advances in psychology, engineering and computer infrastructure will add significant capability in the training device developers' quest of achieving high levels of fidelity. In the future very large numbers of real world stimuli will be presented to trainees at affordable costs. Presenting this number to trainees in tasks involving significant amounts of hand-eye coordination (e.g., gunnery training, aircraft operations) has generally been possible in the past only at high cost. Even where the funds have been made available, selective fidelity has still been required because of our inability to define and present all the possible stimuli. Presenting high levels of fidelity to tactical trainees has really not been possible because of the costs required to replicate the numbers of training devices necessary for sound tactical training. A minimalist selective fidelity approach has therefore been used. We believe that will change in the future.

Gagne (4) referred to internal and external conditions of learning. Internal conditions of learning are what the learner brings to the learning situation. These include; learning style, pre-requisite knowledge and skills, motivation and a host of other internal parameters. External conditions of learning are the parts of the learning system outside of the trainee. These are the elements that we as instructional designers choose and arrange to present learning information to the trainee. All instructional matter and the media to deliver it (including training devices) are external conditions of learning.

In our quest to identify the key training stimuli we should start with understanding the trainees' internal conditions of learning. Only then can we select and arrange the external conditions in an optimal fashion. The advances discussed in this paper will all make it easier to focus on the learners and their internal conditions than has been the case in the past. Cognitive learning and performance theories/tools will give us a better perspective on those internal conditions and thus the stimuli that will need to be presented. Technological advances will make it easier and more affordable to simulate the stimuli that we decide the learner needs. That is not to say that the concept of selective fidelity will disappear. The nature of our selectivity will change. It will then be centered around the trainees' ability to absorb stimuli rather than other factors.

REFERENCES

1. Fink, C.D., & Shriver, E.L. Simulator for Maintenance Training: Some Issues, Problems and Areas for Future Research. AFHRL-TR-78-27, AD-A060 088, July 1978.

2. Alluisi, E.A. The development of technology for collective training: SIMNET a case history. Human Factors, 33, 343-362, 1991.

3. Miller, L.A. et al Training Device Design Guide: The Use of Training Requirements in Simulator Design. AFHRL TR 76-C-0050, June 1977

4. Beckshi, P.F., Lierman, B.C., Redding, R.E., & Ryder, J.M. Procedural Guide for Integrating Cognitive Methods into ISD Task Analysis. AL-TR-1993-0020 1993.

5. Bliss, J., Monk, M., & Ogborn, J.M. Qualitative Data Analysis for Educational Research: A Guide to the Use of Systematic Networks. London: Croom Helm, 1983.

6. Cooke, N.M., & McDonald, J.E. The application of psychological scaling techniques to knowledge elicitation for knowledge-based system. International Journal of Man-Machine Studies. 26., 533-550, 1987.

7. Johnson, L. & Johnson, N.E. Knowledge elicitation involving teachback interviewing. In A.L. Kidd (Ed.), Knowledge Acquisition for Expert Systems: A Practical Handbook. New York: Plenum Press. 1987.

8. Gagne, R. M. The Conditions of Learning. (4th Ed.) Holt, Rinehart & Winston. 1985.

VOICE RECOGNITION: A REBORN TECHNOLOGY FOR EDUCATION AND TRAINING

Wayne E. Creech
Hughes Training, Inc.
P.O. Box 6171
Arlington, Texas 76005-6171
(817) 695-2038

ABSTRACT

Both the development of the computer and the concept of voice recognition date back prior to World War II. While the advancement of computer technology has been steady since the early 1940's, it has boomed since the late 1970's. Conversely, the progress of voice recognition has not been well received by the scientific and technical communities until the last decade. A key reason has been the degree of accuracy with a voice input compared to that of a keyboard or manual input. Ironically, it is the manual interface that has become an obstacle for humans to handle the data entry and systems control functions. The need for an improved interface between the information systems and their users is a prime factor in the current technology research efforts. Voice recognition offers a potential for a more user friendly interface and is the object of renewed interest in both military and civilian communities. This paper will examine the latest advances that have triggered the new interest in this technology, current applications of voice recognition systems, and explore the development of a new application. The structure will be in five parts. Part One is an introduction that defines voice recognition and terms, presents an illustration of a generic voice recognition system, and describes the categories of voice recognition systems. Part Two discusses the current systems' capabilities, identifies the constraints that presently prevent the development of the ideal system, and describes the most popular speech recognition model. Parts Three and Four discuss the potential training application of voice recognition in business, industrial, military, and education communities. Part Five describes the creation of a training application using a developer's kit.

BIOGRAPHY

Wayne Creech is a project leader for Independent Research and Development (IR&D) activities for HTI. He has directed military training porgrams in the Air Force, international training programs in Saudi Arabia and Indonesia, and has held a variety of positions on several U.S. military training contracts while fulfilling requirements for a PhD in Applied Technology, Training and Development at the University of North Texas.

VOICE RECOGNITION: A REBORN TECHNOLOGY FOR EDUCATION AND TRAINING

Wayne E. Creech
Hughes Training, Inc.
Arlington, Texas

INTRODUCTION

Voice recognition is "the technology by which sound, word or phrases spoken by humans are converted into electrical signals, and these signals are transformed into coding patterns to which meaning has been assigned" (Adams, 1990). More simply stated, it is the "process of automatically identifying spoken words" (Foster 93). It is a technology that is enjoying a rebirth from the disappointments of earlier efforts and is having a dynamic increase in scientific interest and consumer applications. The terms "voice recognition" and "speech recognition" are used interchangeably.

System Categories

Voice recognition systems are divided into the categories of speaker dependent and speaker independent. The dependent speech systems must be trained by the speaker. Key elements in the training include (Foster, 93):

- vocabulary choice
- training the recognizer to adapt to the speaker's characteristics
- repetition of each vocabulary word
- training the recognizer in the proper environment

While the speaker dependent systems are successful in many applications, they have several problems (Lee, 89):

- inconvenience of the training session
- processing time after the training session
- involvement of multiple speakers
- additional storage requirements for separate speaker parameters
- variation in speaker voices due to stress, fatigue, or illness

The independent system does not have to be trained to adapt to the speaker's voice, however, the recognizer is trained from a large speech database prior to being used. Approximately 1000 speakers may be needed to reflect a balance of female and male speakers with various dialects (Foster, 93). Studies are currently being conducted on using books on tape as the training data for the speech modeling (Doulianne et al, 94) and (Kenny et al, 94).

Both dependent and independent systems may use discrete or continuous word recognition. Discrete recognition uses a pause of 250 milliseconds to separate each word, whereas continuous recognition has less than 50 milliseconds of silence between words in a series (Foster, 93).

Continuous recognition is more complex due to three factors (Lee, 1989):

1. Word boundaries are unclear since the speaker does not pause between words.

2. Co-articulatory effects result from natural speech, i.e., blending of contiguous words such as "I'm for I am, or "who ch doin," for "what are you doing?"

3. Content words that use nouns, verbs, and adjectives are emphasized while function words like articles and prepositions are not articulated well.

To try to resolve the complexity problems, continuous speech systems are constrained by content-related words called a grammar or syntax. The syntax is related to the type of application, such as inspecting a car on an assembly line or following a troubleshooting flow chart. By using a syntax, the system does not have to consider the entire vocabulary at every decision point (Lee, 89).

GENERIC SPEECH SYSTEM

In both independent and dependent systems, a user speaks into a microphone, normally mounted on a headset. The headset is plugged into a host computer and interfaces with a speech recognition board which has a digital signal processor that converts the voice signal (analog) into digital signals.

converts the voice signal (analog) into digital signals. The digital signals are then analyzed and converted to text or decoded as a command. The processor and analyzer work from statistical and mathematical algorithms that recognize speech blocks, group them into words, and match them against models stored in the program. An illustration of a generic speech system is shown in Figure 1.

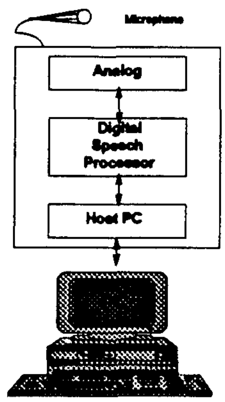


Figure 1: Generic Speech System

CURRENT CAPABILITIES

The technology of voice recognition is being called "revitalized" or "reborn" because of its early rise, downfall, and newly emerged popularity. Like the computer, speech recognition is a concept that predates WWII. There is some documented research in voice technology that dates back to the 1700s, one of the most famous being a talking machine by Wolfgang von Kempelen in 1791 (Nelson, 84). Research continued in the 1800s and early 1900s with the invention of an electronic speech synthesizer in 1936. The technology struggled in the '50s and '60s but did not get its new life until the personal computer became almost as common as the second TV set. Four pivotal reasons for this include: (1) The explosive development of the microchip; (Thyfault, 94) (2) the nesting place provided by the personal computer; (3) the need for a hands-free input; (Adams,90) and (4) the accuracy of a voice compared to a manual input. Item #4 is still one that puts constraints on developing the ultimate system; one that can understand fluent, conversational speech with an unrestricted vocabulary, from most any speaker. While today's systems do not have an unlimited vocabulary, they have taken a gigantic step from approximately 1000 to 50,000 words in just five years. The progress was not as rapid as that envisioned by an elite study group formed by the Advanced Research Projects Agency of the Office of the Secretary of defense in 1970. Their report, submitted in 1972, stated that a speech recognition system should meet the following requirements by 1976 (Newell et al, 73):

- accept continuous speech from many speakers of the American dialect in a quiet room over a good quality microphone
- allow slight tuning of the system per speaker
- require only natural adaptation by the user
- permit a selected vocabulary of 1,000 words
- accept a task like data management
- function from a simple psychological model of the user
- tolerate less than 10% semantic error
- operate in real time

While modern systems meet or exceed these specifications, the study group was on target; their "specs" are as valid today as they were in 1972.

Identification of Constraints

In 1989, the systems that accurately performed voice recognition did so because they operated with the following constraints (Lee, 89):

(1) speaker dependence
(2) isolated words
(3) small vocabulary
(4) constrained grammar

With the exception of a small vocabulary, these constraints are still true today for speaker-dependent systems. For speaker-independent systems, the grammar or syntax constraint is still valid, however, by carefully developing the syntax, the constraint can be greatly minimized. Syntax development is discussed later.

SPEECH RECOGNITION MODEL

A review of current literature shows that the most popular algorithm for developing state-of-the-art systems is the Hidden Markov Model (HMM) (Lee, 89). The HMM sets up a series of states capable of generating outputs. States are analogous to blocks on a flow chart. When the action at a state is completed, the recognizer is ready to move to the next state. When generating a word, each state emits an output until the entire word is out. Movement between states is called transition. A path is a sequence of states. Recognition is achieved by computing the path lengths with respect to the HMM. The word with the shortest path to the HMM is provided to the recognizer as an output (Foster, 93). The model itself is not apparent to the recognizer, so it is called "hidden" (Parsons, 87). Figure 2 represents a HMM with three states.

The HMM can be applied to set a set of states representing the complete vocabulary to recognize continuous speech. The path search is expanded to a language model. The word models are formed into a network that frames the language model and depicts the grammatical constructs or syntax. Figure 3 illustrates a basic language model which allows three phrases to be said with each subject (Foster, 93).

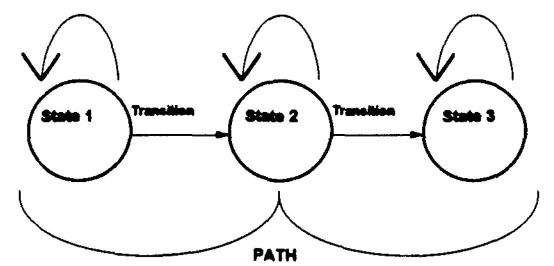


Figure 2:
Hidden Markov Model for Dependent Speech

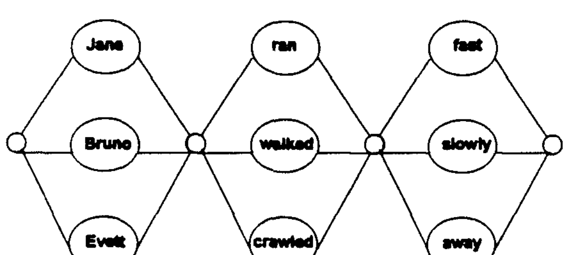


Figure 3: Language Model for Continuous Speech

APPLICATIONS FOR TRAINING

Creating a training application with voice recognition is similar to other uses of technology; the developer must understand the objectives of the training program and use the technology to aid in learning how the objectives can be accomplished. For example, Hughes Training, Inc. recently developed a scenario for a commander in a command post to direct the action in a simulated battle. Radio transmissions to tank commanders involved in battle were simulated through voice recognition; the system recognized call signs and cued an appropriate recorded response from the called commander. The objective was to train the battlefield commander to direct the battle. Voice recognition was embedded in the simulated transmissions and appeared no different to the user than an ordinary radio set. Some future uses for the battlefield involve the Army's Landwarrior uniform which allows a platoon leader to use voice recognition to access data in a portable computer. For air combat, the Air Force is currently studying the use of voice commands for F-16s.

When the scenario is moved from combat training to business or an industrial setting, voice recognition technology is already in place for many users. Receiving operators for Compaq unpack, track returns, credit customers, and order parts simultaneously through voice recognition data entries. Mechanics at some Air Force bases use a headset attached to a computer to state commands like "enter diameter"(Thyfault, 94). While the voice technology is more conspicuous in these environments, it is only a tool to facilitate a quality control task. In this instance, the training tool and the task tool are the same. This same concept applies to doctors who create digital medical records by speaking into a phone attached to a PC, lawyers who dictate letters, and Wall Street traders who tell a workstation to buy or sell shares. Disabled persons who can speak are also finding voice recognition provides a means of overcoming their disability, but the largest applications have come from telephone companies whose systems recognize miscellaneous commands like "collect" and "person to person" (Filipczak, 93). SPRINT is one of the more visible companies with commercials that show Candice Bergen telling her card to "call travel agent".

tool to accomplish the task and not a focal point for training in these areas. One may ask then, "Are there applications that provide the cognitive side of training?" The answer is very few. Most systems come designed for a specific purpose, i., e., medical, word processing, spreadsheets, or have to be developed for the user's requirements. Speech Systems, Inc. has an instructional program designed for air traffic controllers to learn terminology, and the same company is considering developing a program that provides lessons for learning Japanese. Verbex provides a speaker-dependent system which Wesson International uses to build a PC-based simulation system for air traffic controllers. while this sytem is speaker-dependent, it is different from other dependent systems observed in that it uses continuous speech.

What about public schools, an area overdue for the advantages in technology? A review of 246 articles did not reveal one item related to an educational application. Some school systems are using a program called SpeechViewer by IBM to assist with speech therapy. It is composed of modules that help develop an awareness of speech dimensions such as pitch, loudness, and patterning. Telephone interviews with several Texas school districts indicated there were no plans to include voice recognition in the curricula. However, their plans may change; a Waco-based company called Creative Education Institute is considering using voice recognition in a program it has designed for students with learning disabilities called Essential Learning Systems (ELS). The ELS uses a program of sight, sound and tactile stimulation-response techniques that has been successful with at-risk students in Texas and several southern states.

CREATION OF AN APPLICATION

Development Kit

While a speaker-dependent application may only require adapting to the speaker's voice and adding vocabulary, a voice recognition development kit may be required to create a training application with continuous speech. It consists of the recognition board that is installed in a computer, (usually a PC), a headset with microphone, associated software and technical manuals. Some vendors will provide training in how to use their systems for development and assist with technical support for a specified time. Unless the developer has installation and computer programming skills, a software engineer (computer programmer) is required to create an application for continuous speech. An ideal team is an instructional designer, a software engineer, and a subject matter expert. An exception to the programming requirement was noted in the Verbex/Wesson conbination. According to Wesson, only the grammar (syntax) had to be entered into the recognizer.

Microsoft is working rapidly to make the developer's job easier by making an interface that will allow voice recognition in any application. AT&T has recently released a toolkit that enables developers to integrate voice recognition with telephone applications (Thyfault, 94). The PE400 development kit from Speech Systems requires extensive programming.

Hardware Requirements

In addition to the development kit with the 16-bit voice recognition board, microphone, headset and manuals, a 486 PC with 33 Mhz processing speed, and 8 megabytes of RAM are recommended. The recognition board usually has its own memory so an extra-large hard drive is not required. If speech outputs are desired, a sound card must also be installed.

Methodology

Syntax Development. A critical factor in developing a voice recognition application for continuous speech is the design and development of the syntax. The developer must determine the trade-off between the flexibility of the syntax and the performance of recognition. More flexibility translates to more options of expressions which increase the complexity of the syntax. When complexity is increased, recognition performance starts to decrease. The decoding process takes longer, and there are more possibilities for error. According to Speech Systems, Inc., "the optimum design for a speech-based application is one that is both suitably constrained, so as to allow good recognition performance, and also suitably robust, so as to allow the user to speak what is necessary in a natural manner". Deciding upon the level of complexity is an issue that should be made between the developer, the subject matter expert, and the user. Successful use of the application may be the result of convincing the user to restrict spoken expressions. For example, "Power switch on" might suffice for any number of machines including a computer, instead of variations like "Turn on the power switch," or "Switch on the power,". While some users such as medical professionals may require more description possibilities, the developer should try to get the user to determine the essential expressions and agree to use them. A script is indispensable when finalizing the wording. Figure 4 outlines a simple script used in the HTI application for the war simulation.

SITREP SEQUENCE

Command: ALPHA 16, this BRAVO 14, over
Computer response: This is ALPHA 16, over
Command: This is BRAVO 14, SITREP, over
Computer response:
This is ALPHA 16, SITREP
Defending in initial positions
Under attack by estimated company size forced
3 enemy tanks destroyed, 2 tanks mobility kill
2 M1 tanks destroyed, 3 WIA
Continuing to defend, over

Command: This is BRAVO 14, roger, out

UNIT MOVE SEQUENCE

Command: CHARLIE 16, this is BRAVO 14, over
Computer response: This is CHARLIE 16, over
Command:
This is BRAVO 14, move to phase line orange, over
Computer response:
This is CHARLIE 16, WILCO, OUT
(Note: Icons of tank units should start moving now)

Figure 4: Script for Training Scenario

UNIT SPOT REPORT SEQUENCE

Command: DELTA 16, this is BRAVO 14, over
Computer response: This is DELTA 16, over
Command:
This BRAVO 14, give me your latest SPOT REPORT, over
Computer response:
This is DELTA 16, SPOT REPORT
DELTA 16
4 T80 tanks defensive positions
250822May94
Directing artillery fire, over

Command: This is BRAVO 14, out

ARTILLERY FIRE REQUEST SEQUENCE

Command:
ECHO 16, this is BRAVO 14, suppress AB6432, over

Computer response:
This is ECHO 16, suppress AB6432, authenticate TANGO HOTEL, over

Command:
This is BRAVO 14, I authenticate FOXTROT, out
(Note: Artillery fire should commence on the screen at the specified coordinates.)

Figure 4 (cont'd.): Script for Training Scenario

When developing a syntax, a helpful step is to group words that have a similar meaning or comparable in use. These groups are called syntax categories and consist of word and phase-structure categories. The word category is a list of single words which are identified by the symbol ==. Phase-structure categories are grouping of words, phrases, and/or category names, and identified by the symbol >. Others include the S--> which is the sentence rule definition symbol, {} brackets, and | vertical bars. These and other symbols are used by the system's compiler to accomplish the recognition. Figure 5 provides the actual syntax structure developed for the training scenario. Note the use of the sentence rule, word and phase-structure categories. Better recognition was achieved when the syntax was restricted to an almost verbatim use of the script.

```
ALPHA+ == alpha charlie delta
UNIT+ --> ALPHA+ one six
REPORT+ --> SPOT REPORT ' SITREP
COORDINATES+ --> ALPHA BRAVO six four three two
        ALPHA CHARLIE four five seven three
SOURCE+ --> (this is) BRAVO one four
S --> UNIT+ SOURCE+ over
S --> SOURCE+ (give me your latest) REPORT+ over
S --> SOURCE+ move to phase line orange over
S --> SOURCE+ roger out
S --> SOURCE+ I authenticate FOXTROT out
S --> echo one six SOURCE+ suppress
        COORDINATES+ over
```

Figure 5: Syntax Structure

If the user wants more flexibility of expression, techniques called recursion and iteration should be considered, but must be used with caution because they can cause a significant increase in the complexity of the syntax. Recursion allows a series of alternate phrases to be generated from a single phrase. The developer should be alert to ensure the alternate phrases are actually needed as overuse of recursion will result in many more alternate structures than required. A large number of alternates will reduce recognition speed and accuracy. Iteration is similar to recursion but specifies a phrase that can be repeated any number of times. Like recursion, iteration can generate many more alternatives than needed if the developer is not careful.

Response Recording. In the script shown in Figure 3, the computer responses were recorded using a ProAudio Spectrum sound card. When voice recognition occurs, the appropriate response is cued. A RESPONSE file has to be created to tell the speech recognition card where to find the responses.

Creation of the Application. Visual C++ AppWizard was used to generate the prototype code while tying the dynamic link library of the development kit to the PE400 speech card. To allow the output that causes movement of tanks or a sequence of firing, a serial port communication support was added to a silicon graphics machine. The output from the serial port caused the movement of tank icons or a firing sequence.

Demonstration of the Application. The application is demonstrated by assuming the role of the field commander and calling one of the call signs, i.e., Alpha 16. A correct recognition provides a response with the call sign. The called party is asked to provide a report or to accomplish an action. If the response is a report, recognition should take place to use a pre-recorded report. If an action is requested, a recorded response should be cued followed by an output to the silicon graphics interface. The result is either flashed on the screen to simulate tanks firing or icons moving to simulate tank movement.

SUMMARY

Within the last ten years there has been a rapid increase in the interest and development of voice recognition, but only in the last five years have business and industry started to recognize its potential as a labor– and time–saving instrument. As a training medium, it has rarely been used. Some applications for the military were created in the 80's with disappointing results. With today's systems achieving greater speed and accuracy, the Army and Air Force appear ready to give this technology another try. While speaker–dependent systems are proving to be successful and useful in many applications, it is the continuous speech capability that is most desirable for training applications. The technology is just now reaching a point where the speed and accuracy are feasible for speaker independence, but these applications presently require a significant development effort. With companies like Microsoft creating more user–friendly development kits, it is likely there will be a new trend in training applications; the addition of voice recognition as part of the instructional medium. It is a natural extension of computer–based training as well as simulator devices.

When choosing a recognition system, it may be useful to refer to a checklist of items developed from the Advanced Research Projects Agency's study group. They called it "Considerations for a Speech–understanding System," and many of the items are still applicable (Newell et al, 73):

1. What sort of speech? (Isolated words? Continuous speech?)

2. How many speakers? (One? Small set? Open population?)
3. What sort of speakers? (Male? Female? Child? Cooperative? Casual? Playful?)
4. What sort of auditory environment? (Quiet room? Public place?)
5. Over what sort of communication system? (High quality microphone? Telephone?)
6. How much training of the system? (Natural adaptation? Elaborate?)
7. How large and free a vocabulary? (50? 200? 1,000? 10,000? Preselected? Selective rejection? Free?)
8. What sort of language? (Fixed phrase? Artificial language? Free English? Adaptable to user?)
9. What task is to be performed? (Fixed response? Highly constrained?
10. What is known psychologically about the user? (Nothing? Interests? Current knowledge? Psychological model for responding?)
11. How sophisticated is the conversational dialogue? (Task response only? Ask for repetitions? Explain language? Discuss communication?)
12. What kinds of errors can be tolerated? (None <.1%? Not inconvenience user? <10% High rates tolerable? >20%?)
13. How large a memory is available? (1 -1000 megabytes?)
14. How sophisticated is the organization (of the speech processing program?) (Simple program? Discrete levels? Multiprocessing? Parallel processing? Unidirectional processing? Feedback?)
15. What should be the cost?

REFERENCES

Adams, Russ. Sources of Automatic Identification and Data Collection, Van Nostrand Reinhold, New York, NY 1990.

Doulianne, G., Kenny, P., Lennig, M., O'Shaughnessy, D., Mermelstein, P., Books on Tape as Training Data for Continuous Speech, Speech Communication Journal, Feb, 1994, Volume 14, Issue 1, pp. 61-70.

Filipczak, Bob. Adaptive Technology For The Disabled, Training, pp 23-29, March, 1993

Foster, Peter and Schalk, Thomas B. Speech Recognition, The Complete Practical Reference Guide, Telecom Library, Inc., NY, NY, 1993.

IBM National Support Center for Persons with Disabilities, P.O. Box 2150, Atlanta, GA. 1993

Kenny, P.; Doulianne, G; Garudadri, H.; Trudelle, G.; Hollan, R.; Lennig, M.; O'Shaughnesssy, D. Experiments in Continuous Speech Recognition Using Books on Tape, Speech Communication Journal, February, 1994, Volume 14, Issue 1, pp. 40-60.

Lee, Kai Fu Automatic Speech Recognition: The Development of the SPHINX System, Kluwer Academic Publishers, Norwell, MA 1989.

Morgan, Nelson. Talking Chips, McGraw-Hill BookCompany, NY, NY 1984.

Newell, Allen; Barnett, Jeffrey; Forgie, James W; Green, Cordell; Klatt, Dennis; Licklider, J. C. R.; Munson, John; Reddy, D. Raj; Woods, William A. Speech Understanding Systems - Final Report of a Study Group, North -Holland Publish Company, Amsterdam, 1973.

Speech Systems, Inc. Phonetic Engine 400 System Development Kit, 2945 CenterGreen Court South, Boulder, CO 1993

Thyfault, Mary E. The Power of Voice, Information Week, pp 39-46, May 9, 1994.

COMPUTER-BASED ENGLISH LANGUAGE TRAINING FOR THE ROYAL SAUDI NAVAL FORCES

Katharine C. Golas, Ph.D.
Southwest Research Institute

Ronald C. Fredrickson
Margery A. Negri
Naval Education and Training Security Assistance Field Activity

ABSTRACT

The Defense Language Institute English Language Center (DLIELC) is responsible for the American Language Program. DLIELC training materials are used in large-group classroom and individualized language laboratory instruction. Materials may include printed texts for students and instructors, lesson audio tapes, book quizzes, performance tests, and training aids. With recent advances in training and speech recognition technologies, it is now possible to augment such materials with interactive computer-based exercises that use multimedia and voice input to teach English as a second language more effectively. Interactive training that combines audio with full-motion video, still photos, and graphic or animated visual cues has been shown to increase learner motivation by actively involving learners and providing individualized feedback and remediation.

This paper describes a program in which speech recognition technology has been combined with multimedia scenarios that simulate real-life situations and draw the learner into active use of the language. Using speech recognition allows students to improve their speaking skills by requiring them to repeat words and phrases until they are proficient. The system recognizes over 50 words and phrases. The system is currently being evaluated in Saudi Arabia.

ABOUT THE AUTHORS

Dr. Katharine C. Golas is manager of the Instructional Systems Section at Southwest Research Institute. She began her career in Instructional Systems Development (ISD) in 1977, by using the Interservice Procedures for ISD Model to develop print-based exportable job training packages. During the past 17 years, she has directed over 75 ISD projects, including 20 interactive videodisc projects and 10 DVI® projects. She is currently directing research and development efforts using advanced multimedia training technologies. In 1992, she led a project team to redesign the Air Force ISD model and methodology. She has a Ph.D. and M.A. in Instructional Systems from Florida State University. **Address:** Southwest Research Institute, 6220 Culebra Road, San Antonio, Texas 78238. **Telephone:** (210) 522-2094.

Ronald C. Fredrickson has been in military and government service for 35 years, the last 17 years with the U.S. Navy. He has held positions in electronic warfare training design and management, and coordination of international training programs for Central Command and Pacific Command countries. For the past nine years he has been the Training Program Manager for the Royal Saudi Naval Forces efforts in CONUS and in-kingdom under the aegis of NETSAFA. He received the Navy Meritorious Civilian Service Award and a CNO Certificate of Appreciation for significant contributions in support of Desert Shield/Desert Storm. He has a B.S. in Business Administration and has completed many DoD defense and international program management courses.

Margery A. Negri has been with the Department of the Navy for 14 years. She has coordinated training programs for the Royal Saudi Naval Forces for over six years, with four years as the Specialization Schools Training Program Manager, and over two years as the Deputy of the Saudi Training Program Department at NETSAFA. She has an M.A. in Public Administration and Management.

COMPUTER-BASED ENGLISH LANGUAGE TRAINING FOR THE ROYAL SAUDI NAVAL FORCES

Katharine C. Golas, Ph.D.
Southwest Research Institute

Ronald C. Fredrickson
Margery A. Negri
Naval Education and Training Security Assistance Field Activity

BACKGROUND

The Royal Saudi Naval Forces (RSNF) use the Defense Language Institute (DLI) American Language Course (ALC) materials to teach English to all Navy personnel. The DLI course consists of classroom instruction and audio lab exercises. A prototype CBT English language training system was developed to support Book 10 of the DLI course. This language training system has also been designed with sensitivity toward some cultural factors that are unique to the client.

TRAINING NEEDS

One of the most critical training needs today in the RSNF is the need for personnel to learn to speak English. All technical manuals and instructions given to personnel aboard ships are in English. RSNF students often come to the U.S. for advanced aviation and maintenance training. Although students usually receive an average of six years of instruction in English in grade school and high school, data indicates that the average student has a very low English comprehension level (ECL) upon entering the RSNF. Once they have completed basic training, the students go to one of several English language schools for about one year.

DESCRIPTION OF DLIALC

The American Language Course (ALC) is taught at various RSNF schools and the Technical Institute for Naval Studies in Saudi Arabia. The ALC consists of 34 instructional modules (Books 1-34) for teaching English as a Second or Foreign Language, and is designed so that one book builds on the previous book to further language learning and acquisition. The average student takes approximately one year to complete the ALC. The ALC material focuses on four components of the English Language: functions, grammar, skills, and vocabulary.

MULTIMEDIA SUPPLEMENT TO DLI COURSE

The RSNF reported to us that ALC materials do not work very well when the course is taught outside the U.S. because the students in Saudi Arabia do not have the opportunity to put what is learned in the classroom into context. For example, the content of Book 10 includes vending machines and shopping malls. When students go through the ALC at Lackland AFB in Texas, what they learn in the classroom is put into context by their everyday activities. They might use a vending machine or visit a shopping mall and actually use English in context. The few illustrations in the DLI course materials are pen-and-ink line drawings, which do not provide the realism required for the learning to be meaningful.

Research (Al-Juhani, 1991) indicates that computer-based training and videotapes are excellent instructional delivery systems for teaching English as a second language because information can be put into visual context. A full-motion video scenario with audio dialogue can be used as an advance organizer for the student. It ties the objective language elements for each lesson together. By observing these elements used together in an appropriate context, the student should gain a deeper understanding of the elements that he has already learned in the classroom and audio lab.

DESCRIPTION OF TARGET AUDIENCE

The RSNF students in the target audience are between 17 and 31 years old and most have a high-school education. They are not computer-literate. They learn best through memorization and repetition, and tend to be literal, behaviorally-oriented learners. Their motivation is fairly high when they start the English language program but they become bored quickly. The average English comprehension level (ECL) for students entering Book 10 is 30-35 percent.

CULTURAL CONSIDERATIONS

Based on feedback received from the RSNF, the following key cultural issues have been recognized in the CBT:

- There are no women in the CBT.
- There are no churches in the backgrounds, or crosses, bars, nightclubs, or any public place that allows dancing or dispensing of alcohol.
- There are no blatant attempts at humor or mention of the King.

STRATEGIES

General Instructional Strategy

Knowledge engineering techniques were applied during an initial study of the clients' training needs. The conclusion of this initial study was that scenario-based training techniques would be most effective in meeting the clients' cultural, cognitive, and affective requirements. Scenario-based training was accomplished by framing the lesson contents with a continuing story with characters who are present throughout the story. One of the characters is American and the other, more central character is a visitor in the United States from Saudi Arabia. The digital video story shows vocabulary and grammar from Book 10 used in everyday context. The scenarios and stories are presented on a computer screen using compressed digital video and audio. Other course material and text are presented on the same computer. The system is interactive and the student can use a mouse or touch screen to enter information. All exposition, text, instruction, and performance evaluation and testing are done through the same computer system. The students use a headset and microphone to interact through speech recognition hardware and digital audio record and playback. The speech interface gives students practice in vocabulary and pronunciation. At selected points within a lesson, video close-ups of a native speaker's mouth during speech are displayed. The student can use the video to correct the position of his own mouth parts, then speak and compare his pronunciation to that of the native speaker.

Interactive Design Strategies

Over 30 interactive instructional strategies were designed for the program. For example, one strategy employed video vignettes with "cliffhangers." Each of the four lessons in the program ends with an event where the two characters are in trouble. In Lesson 1, they lock their keys in the car and in Lesson 2, they lose each other at a shopping mall. In Lesson 3, they have a flat tire and in Lesson 4 the person they go to the airport to meet is delayed. Since the program is integrated with the classroom instruction (i.e., the student receives Lesson 1 in the classroom, then goes to the CBT for Lesson 1), the idea behind the cliffhangers is to motivate the student to complete the CBT lessons so they can see what happens to the characters.

Another strategy uses dialog practices where the student "speaks for" one of the characters in the video and then can replay his own speech or hear the character repeat the phrase. Games are also developed where students watch a video segment and then select the subject, verb, and object of the verb by speaking directly to the computer. Another strategy is a game where the student takes orders for snacks from three of his classmates. The student then selects the appropriate items from a vending machine and gives each item to the person who asked for it. If he gives the correct item to the classmate, the classmate says "thanks." If he tries to give a classmate something he did not request, the classmate says "I did not order that!" Another strategy shows two events happening in a certain order and asks the student to identify the correct adverb clause of time for one of the events. Word order is practiced by showing the student a photograph or video sequence and asking him to select a sentence that describes the action in the correct order. Another exercise shows a family tree and asks the student to identify how the individuals in the family are related by speaking directly to the computer.

Speech Recognition Strategies

Speech recognition (SR) technology is an interface technology that makes it possible for a computer user to issue commands and instructions to a computer using spoken words alone instead of a keyboard, mouse, or other interface device. Speech recognition (sometimes miscalled "voice recognition") is possible because of advances in digital signal processing (DSP), computer programming, and psycholinguistics. SR depends on the fact that speech utterances have distinct and characteristic acoustic properties. Psycholinguistic research has shown how the human ear and brain process acoustic sounds and many of these processes can be performed by DSP chips hosted on a small personal computer. Many different recognition algorithms have been developed for DSP chips and board sets.

The system selected to support the speech recognition strategy on the CBT was VoiceTools from Dragon Systems, Inc. The following criteria were used to select this SR hardware:

1. Speaker independence
2. Isolated speech (up to 4 seconds long)
3. Recognition accuracy
4. Vocabulary size/customizability (110,000-word vocabulary)
5. Windows compatibility
6. C-language drivers
7. Single-board commercial off-the-shelf product
8. Compatibility with Action Media II, other system boards, and controllers

The English language CBT uses speech recognition subject to the following general principles:

1. The total vocabulary is limited to 300 utterances or less.
2. A maximum of seven utterances from the vocabulary are compared to one another at any time.
3. The speech recognition system is speaker-independent and trained to recognize new speakers with Saudi accents.
4. Users are provided with escape pathways in the event of recognition failure. Human factors considerations led us to the conclusion that no more than three exchanges involving the same word must take place. The programming logic will recognize the constraints and provide for escapes.
5. SR use **never** interferes with or degrades student learning or instruction. SR is used only where it is appropriate and effective.

The English language CBT employs SR technology in two ways:

1. Phonemic stress drills on a subset of Book 10 vocabulary words and phrases.
2. Selection of correct answers to various exercises.

Phonemic stress drills were chosen because of a recommendation from experts in teaching English to non-native speakers. English is a stressed language and many failures of understanding are the result of incorrect syllabic stress. Moreover, stress patterns can easily be detected by isolated-speech, speaker-independent systems. The number of utterances that must be compared at any one time is a simple permutation of the number of syllables in the utterance and can therefore be constrained.

SR is used in approximately half of the instructional strategies, as described previously.

Feedback and Student Data Collection Strategies

If the student gets a correct answer the first time, he is given positive feedback that he was right. If the student makes a mistake on the first attempt, he is told to try again. If the student is wrong twice in a row, he is given the correct answer. The system provides feedback which is generally very positive. The student is never told directly that he is wrong. He is told either to try again, or is given the correct answer. We chose this method of providing feedback to avoid discouraging the student in any way.

The CBT keeps track of student data as follows. For the exercises, data is collected on each input the student has made. We know on which try (first or second) the student was correct or whether the student did not get the correct answer at all. A percentage score is tabulated for each section, and a total score is tabulated for the entire program. A posttest at the end of the program covers the key learning objectives of the CBT supplement. The test contains a selection of interactive exercises and data is collected to show how the student performed on each exercise.

CBT SYSTEM SPECIFICATIONS

The system provides approximately 5 hours of interactive instruction and consists of 30 minutes of full-motion video (all-digital format); 70 minutes of audio, 210 still photographs, and 82 graphics.

Table 1 shows the hardware and software system for the CBT Saudi Language Program.

REPORT OF EVALUATION DATA

Instructional Effectiveness

The CBT supplement was evaluated first at Lackland AFB in San Antonio, Texas, and later in Saudi Arabia.

Table 1. CBT System Specifications

Hardware	CompuAdd 486DX/66 with 20 MB of RAM Quantum 1010is 1.2 Gigabyte SCSI hard disk drive w/controller (Adaptec 1540B) Microtouch 20" touchscreen monitor (Mitsubishi) Diamond Viper Vesa Local Bus VGA card w/2 Meg RAM Logitech MouseMan (3-button) Teac Dual floppy disk drive (3 1/2" and 5 1/4") Intel Actionmedia II Digital Video Accelerator 1/8" male stereo to 1/8" male stereo cable Pro Audio Spectrum 16-bit audio card Altec Lancing Speakers, AC 550 (pair) IBM M-Audio capture and playback adapter (comes w/VoiceTools) Shure SM-10 microphone headset (comes w/Voice Tools) Shure SM-11 microphone and 6-foot adapter cable (female XLR to male stereo) Toshiba double-speed CD ROM drive (optional) 4mm DAT tape backup unit (optional)
Software	DOS 6.0 Windows 3.1 Dragon Systems, Inc. VoiceTools (user version) Novaback Backup and Restore Software version 1.01 (optional)

One Saudi Arabian student enrolled in the DLI American Language Course at Lackland AFB completed the CBT supplement in 4½ hours. The student was enlisted in the Royal Saudi Air Force and had just completed Book 10, receiving a score of 72% on the Book 10 quiz.

General comments based on student performance:

1. The student really enjoyed the video story of Saad and Tom. He asked a lot of questions about Saad and wanted to know where the story was filmed. He enjoyed the reference to buying better coffee for Hassan and the use of the term "Inshallah."
2. The student used the mouse and touchscreen about the same amount of time. He tended to use the touchscreen whenever there was a graphic, and the mouse when text appeared.
3. The student's attention span was very short. He did not seemed bothered at all about the pauses on the program when the computer was searching for a video sequence. He seemed to enjoy having the short wait while the computer was bringing up the information.
4. The student had no trouble with the user interface. Once he had completed Lesson 1, he seemed comfortable with the NEXT button, and he knew to click on the microphone icon before he spoke to the computer for voice recognition and the tape recorder icon to record his voice. The only real problem was that he kept saying the voice recognition words only one time (instead of twice).
5. The student was enthusiastic when he pronounced a word correctly and the computer said "GREAT!" He really wanted to get the answers correct.

The system is currently in Saudi Arabia undergoing a formal evaluation. The purpose of the In-Kingdom evaluation of the CBT supplement to ALC Book 10 is to answer the following questions:

1. What effect does the CBT supplement have on learning the American language? Does the data indicate an **immediate** increase in student ability to speak and understand the American language at the Book 10 level?
2. Does the CBT supplement reduce training time?
3. Does the CBT supplement have an effect on the enthusiasm of the students toward learning the American language? Are the students more motivated to continue with the American Language Course because of the introduction of the CBT?
4. Are some of the instructional design strategies used in the CBT more effective than others at teaching Book 10 content? Instructional strategies include dialog and dialog practices, simulation exercises, speech recognition exercises, grammatical drills, voice record and playback exercises, and the full-motion video "story" of Saad and Tom.
5. What is the instructor's reaction to the use of CBT in English language training?

SUMMARY

A common mistake with many CBT development efforts is that people use the computer to teach things that are best taught using other instructional delivery methods such as print materials (textbook) or classroom instructor-led. The Book 10 CBT supplement does not require the student to **read** a great deal of text off the machine (better achieved with the student text) or **write** a lot. The target audience has minimal word processing and keyboard use skills. The CBT design capitalizes on what the computer does best:

1. Present realistic scenarios through multimedia so the student can learn to speak English in a real-world context.
2. Expose the student to the use of grammatical structures and idioms within a narrative context.
3. Allow the student to recite words and phrases out loud and be able to immediately replay his lines and hear how he did in comparison to the expert.
4. Let the student practice pronunciation and get corrective feedback.
5. Let the student **hear** the correct pronunciation of words and phrases as many times as he needs to.

Data should be available in late 1994 to determine effectiveness of the CBT.

BIBLIOGRAPHY

Albadah, O. A. 1985. *A Plan for Teaching American Culture to Saudi High School Students (Teaching English as a Second Language (TESL), Sociolinguistics, Curriculum Planning).* Volume 47/02-A of Dissertation Abstracts International. University of Kansas.

Aldosari, H. S. 1992. *A Sociolinguistic Study of the Attitudes of Muslim Students, Teachers, and Religious Officials on Learning and Teaching English as a Foreign Language in Saudi Arabia.* Volume 53/05-A of Dissertation Abstracts International. Pennsylvania State University.

Al-Ahaydib, M. 1986. *Teaching English as a Foreign Language in the Intermediate and Secondary Schools of Saudia Arabia: A Diagnostic Study.* Dissertation Abstracts International, A: The Humanities and Social Sciences. 46, 6, December.

Al-Braik, M.S. 1986. *Investigation of the Successful Attributes of English as a Second Language of Saudi Arabian Students Studying in the United States of America.* Volume 47/11-A of Dissertation Abstracts International. Pennsylvania State University.

Al-Juhani, S. O. 1992. *The Effectiveness of Computer-Assisted Instruction in Teaching English as a Foreign Language in Saudi Secondary School.* Volume 52/7 of Dissertation Abstracts International. University of Denver.

Al-Saadat, A. and Afifi, E. 1990. Teaching English via Closed-Circuit Television in a Sex-Segregated Community. *British Journal of Educational Technology.* Vol. 21, No. 3, 175-82.

Ardan, A. A. 1991. *An Exploratory Study of Teaching English in the Saudi Elementary Public Schools.* Vol. 19, No. 3, 253-66.

Christensen, T. 1989. Grammatical Overkill? *English Today.* Vol. 5, 4(20), 38-41.

Magrath, D. 1983. *A "Hands-On" Approach to Teaching English for Science.* Paper presented at the Summer Meeting of the Teachers of English to Speakers of Other Languages. July. Toronto, Canada.

Pack, A. C. Ed. 1978. *Productive Validity of the CELT (Comprehensive English Language Test).* TESL Reporter, Volume 11, No. 3.

Saggaf, A. A. 1981. *An Investigation of the English Program at the Department of English College of Education,* King Abdul-Aziz University, Mecca, Saudi Arabia.

MODEL AND SIMULATION SUBCOMMITTEE

Chair
Jay Anton, Loral Aviation Testbed

Deputy Chair
Paul Hinote, Evans and Sutherland

Members
Art Banman, Concurrent Computer Corp.
Brian F. Goldiez, IST
Bruce Montag, Southwest Research Institute
Dave Powell, Loral Federal Systems Company
Don Mathis, Chief of Naval Education & Training
Dr. Bruce McDonald, Coleman Research Corp.
Dr. Edward Martin, USAF
Earle Denton, E.L. Denton & Associates
Frank Thweatt, STRICOM
Gary Baker, Baker & Associates
George Snyder, Loral Defense Systems
Joanna Alexander, Zombie, Inc.
JoAnne Puglisi, Martin Marietta Information Systems
Joe Brann, Loral Federal Systems Company
Judith Pafford, Hughes Training, Inc.
Ken Kelly, Veda
LTC Dave Bartlett, USMC
LTC Edd Chenoweth, USAF
LTC Jan Drabczuk, STRICOM(PM CAAN)
LTC John Boyd, USMC
Maj. Bob Birmingham, STRICOM PM ACTS
Neale Cosby, Institute for Defense Analyses
Paul Chatelier, White House Office of Science& Technology Policy
Phil Sprinkle, PM TRADE/STRICOM
Ron Schneider, Reflectone, Inc.
Todd Repass, Space & Naval Warfare Systems Command

Section 4
Table of Contents
Modeling and Simulation Papers

TESTING CONFORMANCE FOR DISTRIBUTED INTERACTIVE SIMULATION (DIS) STANDARDS

Amy Vanzant-Hodge, Sandra Cheung, and Scott Smith
Institute for Simulation and Training (IST)
Orlando, Florida, 32826-0544

ABSTRACT

The standards for the interoperability of networked defense simulations, also known as the Distributed Interactive Simulation (DIS) standards, have been prototyped, implemented, and put to the test through interoperability demonstrations at I/ITSEC '92 and I/ITSEC '93 conferences as well as in military programs such as Warbreaker, BFTT, and CCTT. To achieve interoperability between the various DIS systems, all systems must implement the same agreed-to criteria. To ensure that this occurred for the demonstrations at the previous I/ITSEC conferences, the Institute for Simulation and Training (IST) was tasked with testing each system for its level of conformance with the criteria, i.e. parts of the DIS Protocol Data Unit (PDU) draft standard and the Communication Architecture for DIS (CADIS) draft standard. To perform this testing, IST created the DIS Testbed.

This paper describes the DIS Testbed, which consists of hardware equipment, test tools, and test documents, and the test methodologies used for testing. For the I/ITSEC DIS demonstrations a system could be tested in-house at IST, via long-haul connection over phone lines, or on-site at the organization's location. The test methodology used by IST uses a Capabilities Statement filled out for the System Under Test (SUT) and tests the SUT based on its stated capabilities. The tests are outlined in detail in the Test Procedures document. Data from tests is logged with data recording tools and then analyzed to determine if the data is correct. Results from the tests are recorded on a Results Sheet, which is updated for retesting or continuation of tests. A Summary Sheet is filled out when testing is completed and sent to the organization for their review.

ABOUT THE AUTHORS

Amy F. Vanzant-Hodge received B.S. and M.S. degrees in Computer Science from the University of Central Florida in 1983 and 1989 respectively. She is currently Co-Principal Investigator for the Testbed for Research In DIS (TRIDIS) project at the Institute for Simulation and Training. She is currently involved in research for DIS compliance testing, networking, and DIS interoperability.

Sandra E. Cheung obtained her B.S. from Florida Institute of Technology, and Ph.D. from the University of Florida in 1988 and 1993 respectively. Her research interests include performance evaluation, traffic characterization, high speed networks, interconnection networks, and parallel architectures. Since joining IST in 1993, she has been involved in DIS compliance testing, and is currently involved in the analysis of DIS data traces and the characterization of DIS traffic patterns.

Scott H. Smith received an M.S. in Computer Science from the University of Central Florida in 1981. He is currently Project Manager for the TRIDIS project and Program Manager for DIS related activities at IST. His research interests lie in the areas of DIS, CGF, and Constructive Simulations.

TESTING CONFORMANCE FOR DISTRIBUTED INTERACTIVE SIMULATION (DIS) STANDARDS

Amy Vanzant-Hodge, Sandra Cheung, and Scott Smith
Institute for Simulation and Training (IST)
Orlando, Florida, 32826-0544

1. INTRODUCTION

The standards for Distributed Interactive Simulations have been under development through the "Standards For The Interoperability of Defense Simulations" Workshops for the past four years. The first standard to be adopted by the IEEE was the Protocol Data Unit (PDU) standard version 1.0 [1]. Since that time several drafts of version 2 of the PDU standard have been generated and draft standards for Communication Architecture for DIS (CADIS) [2] and Fidelity, Exercise Control and Feedback Requirements (FECFR) have also been created [3]. The latest versions of these drafts are currently going through the IEEE balloting process.

As these draft standards are created, reviewed, stabilized, and balloted, more organizations are using them to implement DIS systems. Because several versions of the drafts exist and these drafts are not always backward compatible, compliance with the different versions must be tested to ensure interoperability. The Institute for Simulation and Training (IST) has had the responsibility of compliance and interoperability testing of systems for the 1992, 1993, and 1994 Interservice/Industry Training Systems and Education Conferences' (I/ITSEC) DIS Demonstrations. Testing is specific to the current version of the DIS PDU draft standard and to the current CADIS draft standard.

1.1 Interoperability In DIS

Interoperability in DIS requires some initial assumptions. First, the version of the standard being used for the "exercise" must be agreed to by all players and this is the version being tested against. Second, inconsistencies in the standard must be resolved for the exercise so that all systems implement the standards the same way. This is key to achieving interoperability. Third, the communications infrastructure (protocols and physical media) must be agreed to by all participants so that this can also be tested. Fourth, a common simulation environment, i.e. terrain database, must be chosen. Once these four points are agreed to, testing can begin.

1.2 The Need For Testing

As stated in the above paragraph, assumptions and agreements must be made in order for systems to participate in a DIS environment. This is because the standards are not specific in many areas. Those areas are left to the interpretation of the developer of the system. Even when the non-specified areas are agreed to, there is no guarantee that the implementations will work together. Testing is needed to insure that the systems will operate in a consistent manner given the standards and the assumptions/agreements for non-specified areas. Testing insures that a system is producing valid DIS PDUs and can receive and interact with valid DIS PDUs. Testing is also used to determine the effects of adverse and erroneous data or conditions on the System Under Test (SUT).

1.3 DIS Testbed Approach

IST has created sets of test tools and test documentation in support of the I/ITSEC demonstrations that have been distributed to organizations being tested as well as being used in the Testbed. Using these tools, systems could be debugged prior to the actual compliance testing. The test documentation, specifically the Test Procedures, indicates which tests each system must pass to be interoperable based on their stated capabilities. Each organization therefore knows ahead of time what tests its systems will be required to pass.

2. DIS TESTBED

2.1 Hardware

The DIS Testbed at IST consists of several types of computers connected by an internal network. The Testbed is configured in such a way that any test system being used for testing can be isolated on a separate network with the SUT. This type of flexibility allows prototyping of new parts of the standard and new types of systems to be added to the Testbed without affecting the rest of the Testbed. Experiments can be conducted with the prototypes/new systems while testing can be conducted unaffected on another part of the Testbed. Long haul connections into the Testbed exist via the use of phone lines and a connection to the Defense Simulation Internet (DSI).

2.1.1 Computer Resources – The IST DIS Testbed currently has several 386 and 486 PCs . Each PC has an Ethernet 10 base 2 (thin coax) interface to the network. The Testbed also has a variety of UNIX based systems; two Sun Sparcs, four Motorola chasses which each contain two 88110 single board computers, and two Silicon Graphics systems which will be used as Stealth 3-D displays. Testing is done with a combination of the PCs and the Motorola's and using the Silicon Graphics to do visual confirmation.

2.1.2 Network Resources – The network in the Testbed consists of thin coax Ethernet. A patch panel is used to allow subnetworks to be connected and disconnected for testing and experiments. To separate machines from the rest of the network, i.e. to isolate a test system and a SUT, the coax cable for that subnet is disconnected from the patch panel.

Recently, the Testbed has been adding new network capabilities, specifically a very flexible network hub. This hub has connections for the thin coax as well as higher speed media such as fiber. Two ports for FDDI, a high speed alternative to Ethernet, and a translation module between Ethernet and FDDI will allow the Testbed to expand and incorporate FDDI devices. The hub also has network management built in so that these various connections can be isolated via software rather than disconnecting a cable and multiple levels of filtering can be done for each connection. Filtering will allow some network traffic to get through while other network traffic is stopped. Using filtering, test systems and a SUT don't have to be physically disconnected from the network. Instead, incoming and outgoing network traffic is limited.

2.1.3 Long Haul Resources – The Testbed currently has two 1-800 numbers which can be used for testing. A BReeze 1000 bridge is used at each end of the phone connection to allow DIS traffic to flow non-stop across the phone line. IST has two BReezes which stay by the Testbed phone lines and four BReezes which can be sent to organizations testing in this manner.

Through the hub mentioned above, a connection to the DSI is being established. A fiber connection allows IST to access the DSI. This fiber is connected to the hub and the Testbed network. Through this connection, IST will be able to exchange network traffic with any organization that is connected to the DSI. This connection will be used for testing and experimentation.

2.2 Test Tools

Testing from any perspective cannot be done without some type of tool to aid in the process. IST has created a set of test tools that perform, record and analyze tests. The test tools were originally created to run on PC machines and have since been ported to other platforms (see Section 5). These tools allow IST to perform the tests as described in the Test Procedures document and to analyze to results.

2.2.1 Data Logger / Playback – The first tool used for testing is the **Data Logger/Playback** system. The Data Logger records packets from the physical network and logs them into a file in either binary or text mode. The recorded file in text mode will take the binary values of the network protocols, including DIS, and translate them into the ASCII or decimal or hexadecimal values. Recording in text mode is very useful for debugging a SUT because the it displays packets in an understandable English format. A recorded binary file can be played back onto the network using the Playback tool. The Playback tool uses the information in the header of the recorded network packet to know when to put the packet back onto the network. Playback is used for generating network traffic and for automation of testing.

2.2.2 Computer Generated Forces (CGF) – The second tool is the **Computer Generated Forces (CGF)** simulator. This CGF has been modified for testing so that it can create many different entities, move them in unusual ways, beam them to locations, beam weapons to locations, move at incredible speeds, etc. The CGF can also accept script files which contain sets of commands used to drive the operation of the CGF. This flexibility is necessary so that the IST CGF can be used consistently and repetitively to force or provoke behavior from SUTs. The CGF is used to create DIS PDUs to preset to the SUT as well as to respond to those generated by the SUT. The planview display of the CGF is used for visual confirmation of SUT behavior.

2.2.3 Scanner Analyzing Tool

The last tool used for compliance testing is the **Scanner**, an off-line packet examination tool. This tool takes a binary file recorded by the Data Logger as input and allows the tester to "scan" back and forth through the packets in the file. The packets are displayed as vertical lines on a horizontal time line based on the time they were recorded and an arrow is used to point at a specific packet on this line. When the arrow lands on a line representing a packet, the contents of the packet are displayed in a text window on the computer screen. The tester can then page up and down through the contents of the packet, through the network headers, and through the DIS data in the packet within this window. The data is displayed in ASCII, decimal and hexadecimal form so that values can be easily checked for correctness.

The Scanner also contains orientation windows which display an entity's roll, pitch, yaw, turret azimuth and gun elevation if the DIS PDU is an Entity State PDU. Future plans for the Scanner include automating as many tests as possible so that testing will be consistent and large numbers of systems can be tested in less time.

3. TEST METHODOLOGY

3.1 Test Documentation

Compliance testing for DIS requires coordination and instruction. In support of the I/ITSEC DIS demonstrations, IST created a set of test documents that contain instructions on how to be tested, how to test, and to how to specify system capabilities [4]. Any organization putting a system through compliance testing needs this information. Other organizations have used the IST documents as a starting point for testing their specific DIS applications. The documents are described in the following sections.

3.1.1 Testing Handbook – The Distributed Interactive Simulation Testing Handbook contains an overall view of the test tools, test plans, test methods and Test policies used for DIS compliance testing of any system. The Test Procedures document is usually included as an appendix to the Handbook.

3.1.2 Capabilities Statement – The beauty of the DIS PDU Standard is its flexibility. This flexibility, however, poses some difficulty in determining exactly what needs to be tested in a system for it to be certified to be compliant. The Capabilities Statement was created for the 1993 I/ITSEC DIS demonstration to document a system's stated DIS capabilities. Based on these, the system may be compliance tested for only those capabilities. This version of the Capabilities Statement asks questions specific to the IEEE 1278 version of the PDU Standard but contains many "Other" categories so that it can accommodate information specific to later versions of the PDU Standard. A system is required to have a Capabilities Statement on file before it will be tested.

3.1.3 Test Procedures – The January 31, 1994, Test Procedures are a full scope document for testing DIS 2.0.3 Draft Standard plus extra tests for design decisions that were made for the I/ITSEC 1993 demonstration, specifically the use of Bit 23 in the Appearance field of the Entity State PDU. This document contains Network Level Tests, PDU Tests, Terrain Orientation Comparison Tests, Appearance Tests, and Interactivity Tests. (For a full description of these test refer to [5]). For each of these categories, the document describes tests for ideal, adverse, and erroneous conditions.

3.1.4 Test Results – There is a methodology for using the Test Procedures to test a system and record the results. The **Test Results Instructions** is an instruction booklet which tells the tester which test to run next, how to run the test, how to data log the test, how to name the logged file, and the criteria for successful completion of the test. The **Test Results Recording Sheet** is used to record the actual results of the test. Ideally, the tester will use the Capabilities Statement to determine in advance which tests need to be completed for the system based on the capabilities of the system. If testing is interrupted or retests need to occur, this is also indicated on the Results Sheet.

3.1.5 Logged Testing Document – For those organizations electing to do compliance testing by submitting data logged files recorded at their site, the Logged Testing Instruction Booklet was created. The Logged Testing Instructions tell organizations how to perform the compliance tests, how to record the data, how to name the files, and which tests to perform, based on system capabilities. The organization will then submit the logged files to IST for analysis of results. The Logged Testing Instructions also inform the organization what test tools are needed to perform and record the tests (IST Data Logger, Computer Generated Forces software, and script and binary files).

3.1.6 Test Status Summary Sheet – After compliance testing is completed for a system, it is necessary to return the results of the tests to the organization as well as to have a summarized version on file for further reference. The Test Status Summary Sheet is a short version of the Test Results Sheet that contains only the indication "Passed or "Not Applicable" and some comments for each test. When this sheet is filled out, a copy of it is given to the organization so that they have a written record of what compliance tests their system has passed.

3.2 Performing Tests

IST currently has four methods of testing available. These methods were developed and have been used for I/ITSEC DIS demonstrations. The first method is for the SUT to be brought in-house to IST and connected to a test network. The second method is for the SUT to be connected to the test network at IST via a long haul connection over the dial-up phone lines. The third method is to use the DIS test tools to run the tests at the SUT facility and log the results. The data logged files are then sent to IST for evaluation. The fourth method is testing on-site at a SUT's facility.

3.2.1 In-House Testing – This method of testing allows the quickest turn around time in testing and data analysis. IST's Testbed lab allows physical access to permit large equipment to be placed next to test systems. Immediate oral and visual feedback as well as operating with a variety of DIS devices are the advantage to in-house testing.

3.2.2 Long Haul Testing – As stated in Section 2.1.3, IST can accommodate testing via phone lines and the DSI. Testing using the BReeze 1000's and the phone lines has proved to be the most popular. Though the BReezes are connected to 1-800 numbers, a separate voice line is needed between IST and the organization in order to conduct the test. As the SUT is put through tests, the network traffic from the tests is logged at IST and then analyzed. Feedback from the analysis is immediate in most cases and usually within a couple hours in other cases.

The most important aspect of this method of testing is the scheduling. Not only is a test time scheduled, but use of the loaner BReezes is also scheduled. BReezes are usually shipped next day delivery, but if there are problems with an address or the organization which had the BReeze previously does not send it on time, it may not arrive at the current organization's facility in time to test. Another problem that limits this method is phone line/switch problems that do not allow network traffic out or into the organizations facility. Proper planning is the key to success.

3.2.3 Logged Testing – The Logged Testing method can be done by any organization which secures a copy of the IST DIS Test Tools. The Logged Testing Instruction Booklet (Section 3.1.5) guides the organization step-by-step in how to conduct the tests and data log the results. The resulting logged files are sent to IST for analysis to be done. Turn around time varies on the level of testing activity within the Testbed, but the organization can schedule to have results by a certain date.

3.2.4 On-Site Testing – This last method of testing involves the movement of test equipment from IST to the site of the SUT. If Long Haul testing cannot be accommodated for systems that cannot be moved, this is the next alternative. IST is working on a more portable test system that will include all the test tools.

3.3 Testing Process

The order of testing, as defined by the Test Procedures, relates to the way a packet is taken off the network and the information in the packet is analyzed in a hierarchical fashion. Network Level tests are performed first to guarantee that the SUT can establish a network connection and send and receive traffic. PDU Level tests come next to verify that the DIS packets are being sent and received with the correct data in the correct fields. Terrain orientation tests and Appearance tests follow for those systems that generate entities. These are used to make sure an entity has the correct orientation and appearance when put through set movements and activities. Last are Interactivity tests which are used to verify interoperability of simulated entities. Finally, once the SUT can handle ideal traffic, errors are introduced into the test data.

The testing process depends on the order of testing defined in the Test Procedures. First a capabilities statement is completed for the SUT. Next, a method of testing is chosen and testing is scheduled. Tests are then performed based on the capabilities for that system. If a SUT fails the tests, then testing must be rescheduled after the problem is fixed. When one level of testing is passed, the SUT proceeds to the next level. When the SUT passes all tests based on its stated capabilities, a Summary Sheet is given to the organization to verify this.

3.3.1 Capabilities Statement – The Capabilities Statement is a crucial addition to the testing process that was not used for the 14th I/ITSEC demonstration. When a Capabilities Statement is filled out for a SUT, it is used to help specify which tests the SUT should be subjected to. If a system claims to have certain capabilities then the system will be tested for those capabilities and no others. This way a system cannot claim to be DIS compliant in areas other than those it is tested for.

For systems that simulate entities, the capabilities of each entity must be listed in detail. For example, a CGF system must have a Capabilities Statement that describes in detail each different entity that the system can simulate.

3.3.2 Hooking up to Test – Once the Capabilities Statement has been completed and the tests selected, a SUT can be tested. The typical setup for compliance testing consists of 2 PCs isolated on a network with the SUT. One PC is used to run a version of the IST CGF software. The other PC is used to record the data from the network using the Data Logger and to analyze the data for correct values using the Scanner. If the SUT is at IST or on-site it is connected via some physical medium directly to the two PCs. If the SUT is connected long haul, a "BReeze 1000" bridge is used on both sides of the phone connection to create a seamless testing network.

3.3.3 Testing and Retest – When the SUT is connected to the test network, testing begins. The order of testing ensures that initial tests must be passed before proceeding. For example, PDU tests cannot begin if the SUT cannot send and receive network traffic. As the SUT is subjected to each level of testing, the results of testing at that level are fed back as soon as possible so that problems may be resolved. If problems can be fixed in a small amount of time, then testing will continue during this test period. If problems are more severe, the organization may have to stop testing and schedule a time for retest when they feel the problem has been resolved.

3.3.4 "Certification" – After compliance testing is completed for a system, it is necessary to return the results of the tests to the organization as well as to have a summarized version on file for further reference. The Test Status Summary Sheet is a short version of the Test Results Sheet that contains only the indication "Passed" or "Not Applicable" and some comments for each test. When this sheet is filled out, a copy of it is given to the organization so that they have a written record of what compliance tests their system has passed.

4. SHORTCOMINGS IN THE TESTBED

The IST DIS Testbed and testing process originated when the first DIS demonstration at I/ITSEC was conceived in 1992. At that time, those organizations who were participating in the DIS Standards Workshops knew that shortcomings in the PDU standard would not allow full interoperability. To insure that systems would interoperate, the concept and process of testing and test procedures for the standard was developed. As the PDU standard has matured and been implemented by more organizations and as DIS demonstrations have continued, the testing process has expanded. Other organizations implemented their versions of test or development tools so that initial on-site verification could take place. Even with this growth of experience with DIS implementations and testing, shortcomings still exist in the standards and in testing.

4.1 Performance in Equipment

One of the first limitations for the IST DIS Testbed was the early decision to use PCs for the test tool platform. The IST DIS Test Tools, specifically the CGF and Data Logger/Playback, were developed as part of a project for low cost CGF prior to conception to use them for testing. The PC platform was a perfect match at the time. For testing, however, the need to have more than 12 entities and the limitations of buffering on the Ethernet card make the PCs inadequate for some of the tests. The same limitations on the network card affect the ability of the Data Logger to capture all DIS packets on the network. For example, high speed aircraft would put out too much traffic for the PC Data Logger to capture. The same test would sometimes have to be repeated three or four times before the data was finally captured.

Another limitation has been on the long-haul equipment. The public phone lines impose a bandwidth limit of approximately of 56 kilobits per second. If this limit is exceeded, the BReeze 1000s have to buffer packets. When the buffers overflow, packets are lost.

4.2 Lack of Tools

At the time of testing for the 1992 and 1993 I/ITSEC demonstrations, the only tools to be used for testing were those developed by IST. Since these tools were PC based, it was easy to find a system to run them on, but the performance of the tools (CGF and Data Logger) was still a factor limiting he type of testing that could be done. This year, the tools are being moved to a faster, more robust platform running (Motorola). This is discussed in detail in the next section.

4.3 Inconsistency In Testing

The last problem that the Testbed faced was inconsistency in testing. Testing conformance to the PDU standard would seem to be straightforward, but there were many areas where bad data was not detected or tests were not run consistently because the expected result was not well defined. The Scanner has the capability to view data inside each network packet. The testing was not automated, however, so that when the Scanner was used to look at data, it was up to the tester to verify values in the fields and to choose the packets to look at. Since only a sample of the data logged file was viewed, not all bad data was caught. Testing in this manner was very time consuming (man power intensive) and boredom on the testers part also contributed to missed results.

5. FUTURE ENHANCEMENTS

In light of the problems discussed in the previous paragraphs, the IST DIS Testbed has been revising the test tools and the testing process to better serve the DIS community.

5.1 More Platforms

The move to improve the Testbed was to take the existing tools and move them to a more powerful platform. The Motorola VME system was chosen because of its flexibility and its ability to provide a high-speed real-time environment for the test tools to run in. The VME system consists of at least two single board computers with RISC processors, one being a motherboard/host and the other being a target. The tools are developed and linked with a real-time operating system on the host and then downloaded to the target board where the executable runs with no interruptions. It has the capability of running extremely fast on the target board. The boards also have a very fast network interface which can accept packets off the network at almost network speeds (10 million bits per second for Ethernet). With this capability, the test tools, specifically the CGF and Data Logger, can be used to test the limits of DIS systems rather that just range of what is specified for an exercise.

In the process of moving the CGF and Data Logger to the Motorola platform, modifications were made so that these tools are almost System V UNIX compatible. With very little effort these tools can be made to be portable to any System V system and can be used for preliminary testing and self testing.

5.2 Automated Testing

The next area of future improvement was to modify the Scanner to be more of an automated test system. Since the Scanner is an analysis tool, it doesn't need to run in real time. The Scanner is being ported to the Motorola system but will run on the host board rather than the target board. The Scanner will be improved in several ways. It will have an X/Motif interface so that it can run on any X platform. Portions of the tests will be automated so that testing is not so manpower intensive. The testing process is being automated so that testing will be consistent for every system tested. All results will be generated by the computer (with hand written comments only where necessary) and the testing process will be menu driven. Default configurations will be set up prior to testing so that all tests for an exercise have the same specifications (ranges, enumerations, etc.). The Scanner will still allow the tester to manually "scan" through a logged file and visually look at the contents of a DIS PDU.

The future of the Scanner analysis tool is for it to be fully automated. An organization that wishes to be tested will acquire a copy of the Scanner, be directed through the tests, step-by-step, be given feedback when a problem arises, and be given the final results when the tests are over. Much of this depends on the stability of the DIS standards and if the draft documents are actually made into standards.

Another tool created to assist with testing is the PDU Editor. This tool will allow a user to create PDUs with any values or edit existing logged files. This tool will specifically be used to create adverse and erroneous data for the tests as well as logged files to play back for tests.

6. CONCLUSIONS

The IST DIS Testbed is a flexible environment that can test the conformance and interoperability of a DIS system. The Testbed has test documents and test tools which accommodate testing and has several methods that a system can be tested. IST's support of the I/ITSEC DIS demonstrations is being used as a means to validate and refine the test tools and test process. Most updates and enhancements to the tools described above are being implemented to support the 1994 I/ITSEC DIS demonstration.

The current testing has been for the DIS PDU Draft Standard 2.0 version 3 and the Communication Architecture for DIS Draft Standard. Until these or future versions of the set of DIS standards are stabilized, the testing for these standards is not stabilized. Testing conformance is straightforward if what is being tested is well defined. There are still many areas of the DIS standards that are not well defined. In these areas assumptions must be made and testing for these areas must be flexible enough to account for these exercise-by-exercise specifications. The IST DIS Testbed has created a test process that works today but must be expanded to meet the future needs of DIS when stability occurs for the various standards. In the meantime, the Testbed (its tools, equipment, procedures) must remain flexible to accommodate the dynamic nature of DIS.

7. REFERENCES

[1] Standard for Information Technology – Protocols for Distributed Interactive Simulation Applications, The Institute of Electrical and Electronics Engineers (IEEE), 1993, IEEE-1278

[2] Draft Standard for Distributed Interactive Simulation – Communication Architecture and Security, Institute for Simulation and Training, University of Central Florida, March 1994, Report IST-CR-94-15

[3] Draft Standard for Distributed Interactive Simulation – Exercise Control and Feedback Requirements, Institute for

Simulation and Training, University of Central Florida, March 1994, Report IST-CR-94-12

[4] "Test Documents for DIS Interpretability," Vanzant-Hodge, A., Institute for Simulation and Training, University of Central Florida, January 1994, Technical Report IST-TR-94-03

[5] (1993). "Computer Generated Forces at the DIS Interpretability Demonstration," Loper, M.L., and Petty, M.D., Proceedings of the Third Conf. on Computer Generated Forces and Behavioral Representation, March 17-19, 1993, Orlando, FL, pp.155-167.

DYNAMIC LATENCY MEASUREMENT USING THE SIMULATOR NETWORK ANALYSIS PROJECT (SNAP)

Richard Barry Bryant	Capt. D. Scott Douglass	Ronald Ewart	Gary Jeff Slutz
WL/FIGD	WL/FIGD	WL/FIGD	EAI Services; Div of Halifax Corp
2180 8th St.	2180 8th St.	2180 8th St.	2180 8th St.
WPAFB, OH 45433	WPAFB, OH 45433	WPAFB, OH 45433	WPAFB, OH 45433

BIOGRAPHIES

Barry Bryant received his bachelor's degree in electrical engineering from Old Dominion University in 1984. He entered the Air Force and served at the Flight Dynamics Directorate's flight simulation facility for five years. He received his masters degree in computer engineering from Wright State University in 1988. Barry has been working as a civil servant in the Flight Dynamics Directorate since 1989. Barry has been involved with real-time simulation computer hardware, simulator I/O systems and simulation networking with the Flight Dynamics Directorate's Simulation Facility since arriving in 1984.

Captain D. Scott Douglass received his bachelor's degree in electrical engineering from Clemson University, SC in 1988. At that time he was also commissioned as an officer in the United States Air Force and stationed at Robin AFB, GA as an airborne avionics project engineer. In 1990, Captain Douglass attended the Air Force Institute of Technology (AFIT) and received his master's degree in computer engineering. Currently Captain Douglass is a simulation systems engineer and is heavily involved in networked simulations.

Ron Ewart received his bachelor's degree in electrical engineering from the University of Akron in 1971. From 1971 until 1990, he provided engineering support, primarily in the area of visual systems, to the Simulator Program Office at Wright Patterson Air Force Base. Since 1990 he has been the Principal Scientist for the Control Integration and Assessment branch, at Wright Labs.

Jeff Slutz holds a bachelor's degree in computer engineering and a master's degree in systems engineering both from Wright State University. Jeff works for EAI Services, a division of Halifax Corp. as an on-site contractor at the Flight Dynamics Directorate's Simulation Facility, Wright Labs, Wright Patterson Air Force Base, Ohio. He has worked at the facility for 15 years and has been a systems/software engineer, the computer systems analyst, and is currently the senior systems engineer for the facility. He is heavily involved in simulator performance evaluation, cue synchronization and timing issues for the facility's multi-processor environment.

ABSTRACT

The purpose of the Simulator Network Analysis Project (SNAP) is to develop networked simulation analysis hardware which measures network delays and simulator accuracies. Latencies introduced by simulations affect the research and training value of the simulator. Critical tasks such as handling qualities evaluations can only be accomplished on high fidelity simulations with very short time delays. In order to be an effective simulation, the pilot's stick input must cause the proper time phased responses from the aircraft's simulated instrumentation, motion base, and visual displays. The introduction of additional time delays between networked simulators, due to line communication links and protocol hardware, reduces the types of tasks which can be accomplished. A thorough understanding of the end-to-end simulation time delays is required to know what types of tasks can be accomplished on the simulation network.

The SNAP hardware and software consists of portable test units which can be located at two or more simulator network nodes. These units have the capability to accurately record and correlate, raw pilot inputs such as stick position, instrumentation, simulation states, network PDUs (Protocol Data Units) and visual display parameters. Each SNAP unit records data at its simulation nodes; the data is time stamped using GPS (Global Positioning System) clocks for subsequent correlation. SNAP provides the unique capability of correlating the inputs of a pilot at one simulation site to the perception of those actions at a second site. SNAP also measures the display attitude and aircraft target position directly from the video going into the pilot's display using an Electronic Visual Display Attitude Sensor (EVDAS).

This paper discusses the development of the SNAP simulator analysis tool as well as experiments and results of the use of the tool on an existing simulator network. Techniques for using SNAP as a simulation verification tool are discussed. Future applications of the tool are proposed.

DYNAMIC LATENCY MEASUREMENT USING THE SIMULATOR NETWORK ANALYSIS PROJECT (SNAP)

Richard Barry Bryant	Capt. D. Scott Douglass	Ronald Ewart	Gary Jeff Slutz
WL/FIGD	WL/FIGD	WL/FIGD	EAI Services; Div of Halifax Corp
2180 8th St.	2180 8th St.	2180 8th St.	2180 8th St.
WPAFB, OH 45433	WPAFB, OH 45433	WPAFB, OH 45433	WPAFB, OH 45433

BACKGROUND

Long haul networked simulation has been rapidly developing for the past several years. It provides the potential for conducting simulations between many players at sites across the country. In order for these large scale simulations to perform meaningful research or training, it is essential that the performance of the simulations be known in terms of network latencies and simulation accuracies. In general, as the latency between two simulations increases, the types of tasks which can be conducted on the simulation are reduced. For example, handling qualities simulations performing a tracking task between two aircraft must faithfully duplicate the aircraft performance with total latencies less than about 150 ms for meaningful training. Beyond-visual-range (BVR) engagements are more forgiving.

Simulation developers have long understood the importance of properly time phasing cues presented to pilots. The simulator instruments, motion system, visual display and HUD must all react to aircraft control inputs just as they do in the actual aircraft. Developers routinely measure the time phasing and accuracy of cues presented to the pilot on single-site simulators. These measurements are not easy to make on long haul networks because the physical separation makes tools such as strip charts and conventional data recording techniques unusable. A network simulation analysis tool is needed so that the performance of long haul networked simulations can be accurately determined. This information is needed so that human factors experts can determine what types of tasks can be done on a particular type of network simulation.

Additional measurements need to be made for long haul networked simulations. Unlike a typical local simulation which runs synchronously, networked simulation delays are not deterministic. Network delays vary as a function of time depending upon number of players on the net and the activity of the simulation. Dead reckoning algorithms are used to determine when Protocol Data Units (PDU) are transmitted on the network. If highly dynamic vehicles such as aircraft in an air-to-air engagement are involved in the task, the rate of packet transmission increases; this may result in increased network latency. Simulator performance measurements must include dynamic latency measurements so that the variations in latency as well as the typical delays can be determined.

THE SNAP PROJECT

The purpose of the Simulator Network Analysis Project (SNAP) is to develop networked simulation analysis hardware which measures network delays and simulator accuracies. The concept is to develop individual simulation measurement units which can be transported to various nodes on a networked simulation. Each unit passively gathers data from its local simulation and accurately time stamps the data as it is gathered. The data from all simulations is subsequently combined for analysis.

SNAP was developed by the Control Integration and Assessment Branch (WL/FIGD), Wright Laboratory at Wright Patterson Air Force Base (WPAFB), Ohio. The project was funded by the Training SPO that is part of the Aeronautical Systems Center also at WPAFB.

Potential delay between two interactive simulations can come from several sources. Each simulator has its own characteristic performance and delays. Each of the boxes on the network also add their own component to total end-to-end time delays. Its easy to see a potential for delays with network interface units, bridges, encryptors/decryptors etc. on both ends of the network. The network itself including packet switching nodes also contributes to the overall delays.

Goals and requirements for SNAP were established based upon Wright Lab's past experience of measuring single site simulations coupled with a desire to understand the detailed performance of interactive long haul simulations. SNAP takes a systems approach toward making the measurements and attempts to determine the simulators' and network performance and relate that to the cues which the pilot perceives. End-to-end time delays are measured between two simulator sites. Latencies can be measured from pilot stick movement at one site until the effects of that control input are perceived by the pilot at the second simulator site.

REQUIREMENTS

The SNAP units needed to be portable so that they could be shipped to various simulator sites for experiments. It was also desirable to use as much commercially available equipment as possible to reduce development costs. The SNAP system is based upon a commercially available Intel based personal computer and uses standard input/output (I/O) cards, Ethernet cards, and a Global Positioning System (GPS) card. Details of the hardware are discussed in the hardware section below.

SNAP requires the ability to monitor basic simulator parameters such as stick position and visual display attitude. These two parameters are of special interest because they typically represent the first pilot stimulus and the last simulator response.

SNAP's accuracy requirement for data correlation between two simulator sites anywhere in the world is 1 ms. Typical simulators have frame rates in the order of 40 Hz. to 60 Hz.; the measurement technique needs to be about an order of magnitude better so as not to affect the data. SNAP's GPS clock has an inherent accuracy of +/- 5 microseconds; however, this accuracy is degraded by the necessary interrupt handlers in SNAP.

One key requirement for SNAP is to be completely passive and utilize zero network bandwidth. This requirement was established to improve the credibility of data measured by SNAP. Unfortunately this requirement eliminates the ability of SNAP to correlate the data in real-time. No data is passed between the units via the simulation network during the measurements. After the measurements are made, the files are transferred via a network or any type of file transfer mechanism.

The capability to measure delays over a complete run is also required. It is important that SNAP be able to continuously determine the simulation delays and variations of delays throughout the experiments.

HARDWARE IMPLEMENTATION

There are two major hardware components that are used to give SNAP its specialized simulation analysis capabilities, each are housed in shock mounted cases so they can be shipped or transported by airlines (see Figure 1).

The first major hardware component is the SNAP computer, which interfaces to the user, hosts the SNAP peripherals, controls the data gathering process, and analyzes the data. The SNAP computer is based on an Intel 80486/50MHz PC ISA bus computer containing 16Mbytes of RAM, a 3.5" floppy drive, and a removable hard drive with 120Mbyte cartridges. These removable cartridges allow gathered data to be secured if it contains sensitive information. The physical structure of the SNAP computer is a 19" rack mounted system with an integrated 10" VGA color monitor and pull-out keyboard.

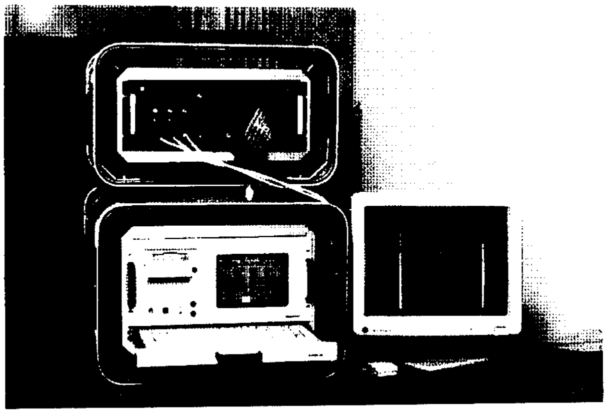


Figure 1. SNAP Photo

The computer's expansion slots contain several off-the-shelf PC cards and an in-house designed and built card (see Figure 2). The off-the-shelf PC cards include a multi-function input/output board, two Ethernet boards and a Global Positioning System (GPS) board as well as the controller card and VGA card. National Instruments' multi-function I/O gives SNAP the capability of outputting two analog signals so that SNAP can drive control inputs for certain tests, inputting eight differential analog signals, and four digital channels (16-bit, 4-bit, 2-bit, and 1-bit digital channels). The Ethernet boards are used to 'sneak' onto Distributed Interactive Simulation (DIS) networks and extract DIS 2.03 protocol data unit (PDU) information. Two Ethernet boards are used so that a single SNAP computer can monitor PDU traffic at two different points within a single simulation site. A global positioning system (GPS), which is accurate to within 5 microseconds, is used to correlate remotely gathered data when two SNAP machines are used. SNAP's GPS is composed of an antenna/receiver, a GPS synchronized timing module, and a PC plug-in card.

The second major hardware component, the Electronic Visual Display Attitude Sensor (EVDAS)[1] board, was designed developed and constructed in-house and interfaces the SNAP computer to the SNAP EVDAS hardware. EVDAS monitors raster video signals going

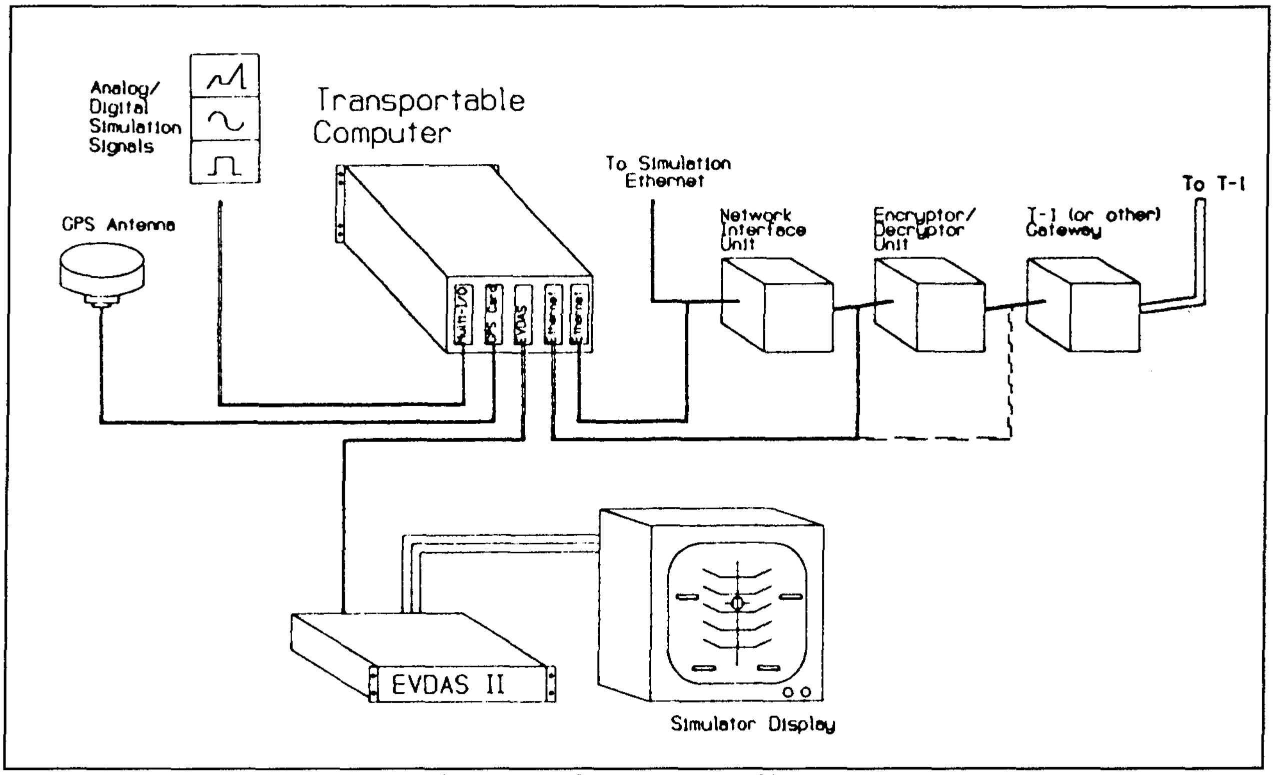


Figure 2. SNAP Hardware Block Diagram

into the pilot's display and determines roll and pitch angles in real time. An analog EVDAS was originally developed at Wright Labs in 1991 for in-house simulator visual system performance measurements and synchronization. A new digital version of EVDAS was developed specifically for SNAP. It has a new capability of centroid tracking and simultaneous pitch/roll calculations so that additional experiments can be done. For example, EVDAS can be set up to measure the angular position of the centroid of an aircraft in the display. EVDAS electronically measures the various parameters during the active portion of the video field. Data is output at the beginning of the video raster's vertical interval. This timing corresponds to the accepted definition of transport delays for visual systems, i.e. when the video field has completed. An interrupt is sent to the computer so that the data can be accurately time stamped prior to being recorded in memory.

EVDAS makes its attitude measurements by generating electronic sensor stripes which are essentially digital timing pulses used to sample video at various points throughout the field of view. These sensors can be monitored visually on a test display using a built-in intensifier function for setup; however, the pilot does not see any sensors on his display. Figure 3 shows an example of how EVDAS determines roll and pitch angles from raster video. Two vertical sensor stripes, one located near each edge of the video are used to sense the presence or absence of sky. EVDAS electronically measures when a certain color is located behind the sensor stripe. In the case of horizon angle, EVDAS 'looks' for blue sky behind each sensor stripe and measures the X and Y coordinate of the intersection of the sky with each stripe. The resulting two X,Y coordinates are used to calculate simultaneous roll and pitch angles. Other sensor shapes are used for other applications. A circular sensor stripe can be used in conjunction with a miniature video camera to determine the angle of an instrument needle directly from the pilot's instrument panel.

SOFTWARE IMPLEMENTATION

The SNAP computer is based on an Intel 80486 ISA bus PC, running a fully preemptive priority based operating system. The basic requirements for the SNAP software are to: provide a user interface to the SNAP computer, record and time stamp data, and perform data reduction and analysis on the recorded data. While DOS has a strong allurement of available libraries for graphical user interfaces and data analysis functions, DOS does not have an acceptable memory or interrupt structure for real-time data processing.

SNAP's operating system is Intel's RMX (iRMX) for Windows. This is a fully preemptive priority based multi tasking operating system which allows one separate and subordinate standard mode DOS task. That DOS task has the look and feel of a normal 286 based computer and can run DOS applications including standard mode Windows. SNAP's DOS task contains the graphical user interface and the data analysis functions. These functions communicate with SNAP's iRMX real-time tasks via shared memory blocks and data files. This merger of DOS and iRMX provides both the benefits of the iRMX real-time operating system, and standard libraries and utilities available to DOS.

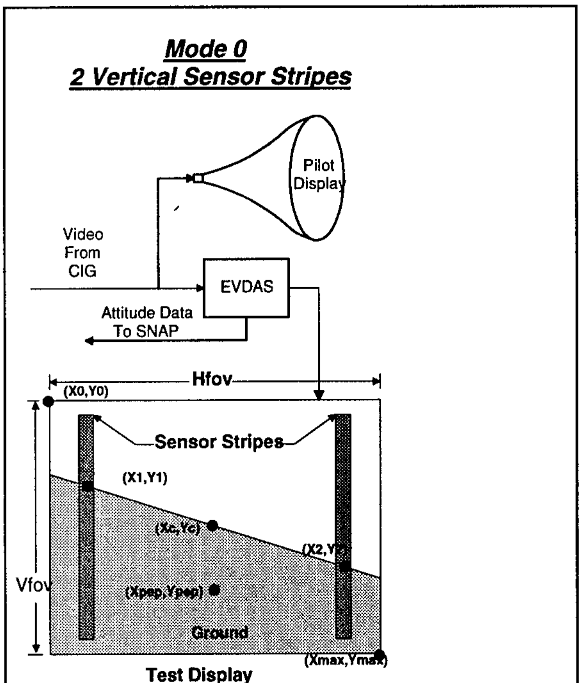
The DOS Graphical User Interface (GUI) allows the user to control SNAP's hardware and software. National Instrument's LabWindows, which provides a user interface library and editor, was used to develop the GUI. The user interface library was used in conjunction with the interface editor to create pull-down menus, dialog boxes, panels and controls, and display graphs and strip charts, all of which are used to display data or receive user input (see Figure 4).

In the configure mode the user can configure and schedule all analog input and output channels, and digital channels as well as indicate on what type of interrupt (internal, external, or EVDAS) those channels are to be read. The user can also configure Ethernet data by indicating what type of filters to place on received PDU packets. EVDAS is also configurable from within this mode. As the user sets the configuration within the DOS GUI, the information is then passed to the RMX operating system for its use.

In the operate mode, commands for starting and stopping runs are given as well as a command for saving data from the last data gathering run. Up to six A/D and D/A channels and all digital channels can be selected and viewed simultaneously in real time. The operate mode also gives the user the option of viewing graphs indicating network activity, to include the number of PDU packets received per second, total number of PDU packets received and detailed information on the last PDU packet received.

The SNAP analysis software is written in "C" and makes use of Lab Window's libraries for its graphical user interface (GUI) and some of its analysis functions. The analysis software is functionally partitioned into two primary sections. Those sections consists of the "replay" section and the "analysis" section.

The replay function of the analysis software allows the user to play back one or more previously recorded files at one of three selectable speeds. It's primary purpose is to allow the user to review a run and to select a portion of the recorded file or files for further analysis. The user can pause the review process at any time and optionally transfer the currently displayed data to the analysis section for a more in-depth look at the data. Replay displays the data in the form of four strip chart traces that are capable of



Points	Screen Coordinates
0	top left
max	bottom right
c	center
pep	pilot eye point reference
1	sensor stripe 1 horizon intersection
2	sensor stripe 2 horizon intersection

$$\text{aspect\_factor} = \frac{Ymax - Y0}{Xmax - X0} \cdot \frac{Hfov}{Vfov}$$

$$\text{theta\_factor} = \frac{\text{vertical degrees}}{\text{pixel}}$$

$$\text{visual roll angle} = \varnothing = \tan^{-1} \frac{(Y2 - Y1)}{((X2 - X1) \cdot \text{aspect\_factor})}$$

$$\text{Pitch angle} = \text{theta\_factor} \cdot \cos \varnothing \cdot \left(Y1 + \frac{(Xc - X1) \cdot (Y2 - Y1)}{X2 - X1} - Ypep \right)$$

Figure 3. EVDAS Concept

displaying variables from one or more files. In the case where variables from different files are being viewed, the correct relative time is maintained by making use of the GPS time stamp in the data files. The time window (the amount of data being presented at one time) is user selectable as well as the amplitude scaling of the strip charts.

The analysis section of the software allows the user to analyze the previously recorded data. Complete files or the portion as selected by the replay function can be chosen for analysis. The data is presented in the form of a four trace plot ,much like an oscilloscope. As with the replay function variables can be selected from different files with the relative time fixed by the GPS time in the files. Once a variable is assigned to a trace, certain mathematical operations can be performed on that variable. The currently available operations are normalization, addition of a scalar, multiplication by a scalar, and inversion. A delay function is also available which calculates the time difference between two traces by shifting one trace to minimize the mean square error between the selected traces. A graphical method to determine the delay between two variables is also provided by the software. In this method the user can "grab" a trace with the mouse and shift it as desired to obtained the best match with another trace. The amount of shift in both the x and y directions is displayed allowing the user to manually determine a time or amplitude difference. Other cursor readouts are provided that display the current position of the cursors and the relative difference between them. This provides a method for the user to make time and amplitude measurements on a displayed variable.

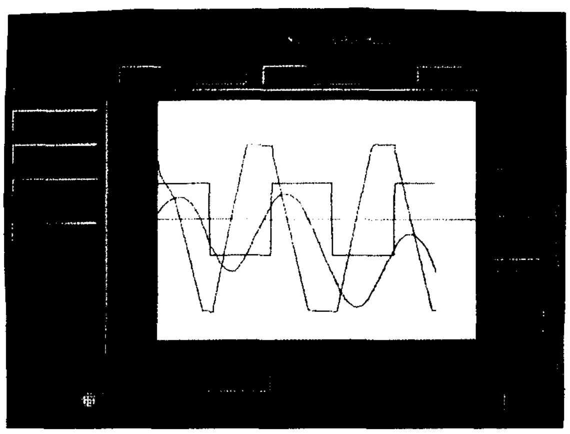


Figure 4. Photo Of GUI

Another important function contained in the SNAP analysis software is the file save function. This function allows the user to save the snap recorded data to a file in an ASCII column oriented format that is suitable for import into a spreadsheet program. The analysis software provides for the case where variables have been recorded at different rates, by creating a common time base (needed by most spreadsheets). This common time base is formed by merging the time bases of all variables. At time points where a particular variable does not have a recorded value, a value for that variable is determined by interpolation.

Future revisions of the analysis software will include a more extensive set of trace mathematical operations. Included will be a set of operations that will allow the user to add, subtract, multiply, and divide two traces as well as use the ratio of traces to determine trigonometric functions. In addition functions will be provided to allow the user to analytically determine the delay in both the time and frequency domains.

iRMX tasks perform the real-time processing for the SNAP computer. The real-time requirements are to respond to external events based on setup information received from the DOS GUI. The external events are: clock interrupts, EVDAS end of video field interrupts, external clock inputs (typically host simulator clock inputs), and receipt of Ethernet data interrupts. The response to an event is to generate a highly accurate time stamp marking the occurrence of the event, and to read and write data based on the data channels scheduled by the user. For each event, an iRMX interrupt task and an iRMX data task are created The interrupt task reads the EVDAS offset timer to generate a microsecond offset from the beginning of the data run and reads inputs and generates outputs based on the GUI generated I/O schedule. The event's data task manages the data in memory and handles the writing of data files at the completion of a run.

Each event's interrupt task has two sections: the interrupt handler and the interrupt task itself. The interrupt handler (Bottom Block of Figure 5) runs at hardware priorities and is uninterruptable. In the SNAP software design, the interrupt handler records the time stamp and exits by transferring control to the lower priority interrupt task (Middle Block of Figure 5). This design has two fundamental effects on the SNAP real-time software performance. The first effect is that time stamp recording takes precedence over all other operations resulting in accurate time stamping of events. The second effect is that the period of time that an uninterruptable interrupt handler is running is absolutely minimized, allowing other interrupt handlers to time stamp other events without delay. The interrupt tasks operate at the highest interruptable task priorities to record data and generate signals as scheduled by the user via the DOS graphical user interface. The only thing that can preempt an event interrupt task is an event interrupt handler. Then the event interrupt task with the highest priority will run until all event interrupt tasks are complete.

The case of Ethernet events is an exception. The Ethernet event does not have an interrupt handler. The interrupt is not serviced directly by a SNAP task but by an iRMX Ethernet job. This job is the data link layer driver for the Ethernet interface. This Ethernet driver signals a SNAP Ethernet task which runs at an interruptable priority higher than the other interrupt tasks. The SNAP Ethernet task implements the IP, UDP protocols. This task performs the same time stamp and data reading functions of the other event interrupt tasks and is only interrupted by the other event interrupt handlers.

Event interrupt tasks complete by sending recorded data to the event data task (Top Block of Figure 5). The event data tasks run at the lowest real-time priority and can only run when all event interrupt handlers and interrupt tasks are complete. The event data task store the data, recorded by the interrupt task, in memory and write the data to disk at the completion of the run. This priority based system yields consistent and accurate time stamps, and the ability to handle surges of interrupts which occur when several events occur simultaneously.

The DOS task runs at the lowest priority (Upper Right Task of Figure 5). When SNAP is not recording

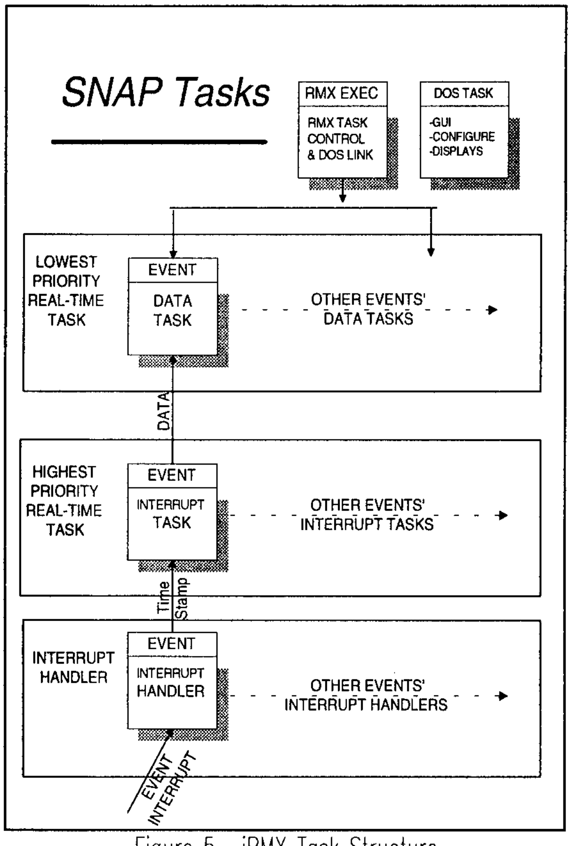


Figure 5. iRMX Task Structure

data, the DOS task is the only SNAP task and has full attention of the CPU. When SNAP is recording data, the DOS GUI which takes user inputs and drives real-time displays only runs when all event interrupt handlers, event interrupt tasks and event data tasks are complete.

VERIFICATION

Several tests were performed to verify SNAP's accuracy. Verification was performed at two levels. The first being iRMX interrupt response and the second being tests of the correlated time stamps.

The purpose of the first test was to verify the real-time performance of the iRMX for Windows operating system and the multi tasking software implementation. In the experiment, a signal generator produced a square wave input to an external interrupt on the National I/O board. The rising edge of the square wave generated the interrupt. The external interrupt, interrupt handler would toggle the state of a digital output and the interrupt task would again toggle the state of the digital output. The relationship between the toggling digital output and the signal generator input to the system showed the latency from interrupt asserted to interrupt handler running, and the latency from interrupt handler completion to interrupt task begin. A typical waveform as seen on the oscilloscope is shown if figure 6.

To get an accurate picture of a fully operational SNAP system, this test was performed with iRMX servicing timer interrupts and generating a waveform on a D/A converter. The DOS task was actively scrolling a graphical plot of the iRMX generated waveform.

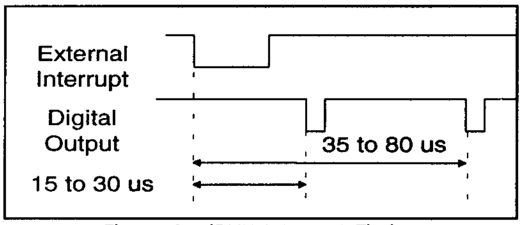


Figure 6. iRMX Interrupt Timing

The first falling edge of the digital output occurred between 15 and 30us after assertion of the interrupt. The second rising edge occurred between 35 and 80us after the interrupt. This test demonstrates that interrupt handler time stamps vary only 15us due to variation in interrupt handler latency. The test also shows that the interrupt task runs within 80us to perform the actual data capture

The second tests involve verifying the correlation of time stamps recorded by two independent SNAP systems recording the same event. For each test, two SNAP computers are setup to time stamp the same external event. Then the time stamps that the two SNAP systems place on the event are compared for accuracy.

Two event sources were tested. The first event tested was an external interrupt input to both SNAP computers. Over ten runs with at least 100 external interrupts per run, the average difference in time stamps of the external interrupt event was 12.6us. The second event was an EVDAS experiment where the same video image was input to each SNAP computer's EVDAS box. The EVDAS events were discrete steps in roll angle. Over ten runs with 150 samples per run, the average difference in time stamps of the roll angle steps was 15.4us. These test verified that two independent SNAP computers could generate GPS correlated time stamps of the same event.

EXPERIMENTS

SNAP provides the ability to measure simulation parameters at key waypoints within the simulation architecture and to drive the simulator inputs with known test signals. Analog control inputs are input directly into SNAP via the A/D's. Simulation state variables are recorded by outputting them from the simulation through D/A's to SNAP A/D's or through a digital interface to the digital input of SNAP. The response of analog queuing devices such as motion systems are determined by connecting instrumentation outputs (accelerometers, rate gyros, etc.) to the SNAP A/D channels. The response of the pilot's visual displays are determined by measuring the roll and pitch angles of the horizon via the Electronic Visual Display Attitude Sensor (EVDAS)[5] interface. The network traffic in the form of PDU's is captured by SNAP via its Ethernet interface. By monitoring the arrival and departure times of PDU's at each site, a statistical picture of network delay as a function of time can be realized. SNAP's GPS clock provides an extremely accurate common time base at each site to properly evaluate these PDU's.

A simulator can be driven by SNAP by connecting the SNAP D/A's to the analog inputs of the simulator. The D/A's can be configured to drive the simulator with a variety of wave forms with varying amplitudes and frequencies.

The setup for a single node pitch experiment is shown in figure 7. The longitudinal stick input is directly connected to one A/D channel, while pitch rate and pitch angle are input to SNAP utilizing simulator D/A's. The pilot's visual display device is connected to SNAP through the EVDAS interface with EVDAS setup to measure the pitch angle of the horizon. The input is provided by SNAP via a D/A. To measure the transient delay under controlled conditions a 0.25 Hz square wave was applied to the stick input. Figure 8 shows the results of this experiment. The delay from the rising edge of the input to the corresponding change in the pitch angle is approximately 230 ms. The transport delay between the true pitch angle and visual display pitch angle can be clearly seen and is in the range of 50 ms. resulting in a total end-to-end delay of about 280 ms.

Figure 9 shows the setup for a typical dual node pitch experiment. A SNAP device was located at each simulator and recorded the data for that simulator stamping it with GPS time. After the experiment was complete the data from the two simulators was analyzed utilizing the analysis software developed for SNAP.

The SNAP device at simulator 1 is setup to record the longitudinal stick input, pitch rate, and the pitch angle through A/D's. At simulator 1, EVDAS is set up to measure the pitch angle of the horizon on the pilots visual display. Outgoing and incoming PDU's are recorded via the SNAP Ethernet interface. The SNAP device at simulator 2 is set up to record the pitch angle of aircraft 1 through a D/A -A/D connection. The display device at simulator 2 is set up to display the out the window view of aircraft 1 based on simulator 2's knowledge of aircraft 1's orientation. This allowed a direct comparison of the pitch angle at simulator 1 with the perceived pitch angle at simulator 2. The EVDAS at simulator 2 is setup to measure the pitch angle of the horizon. SNAP at site 2 is setup to record outgoing and incoming PDU's.

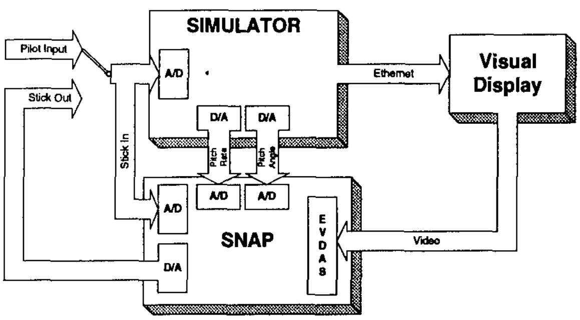


Figure 7. Single Node Experiment

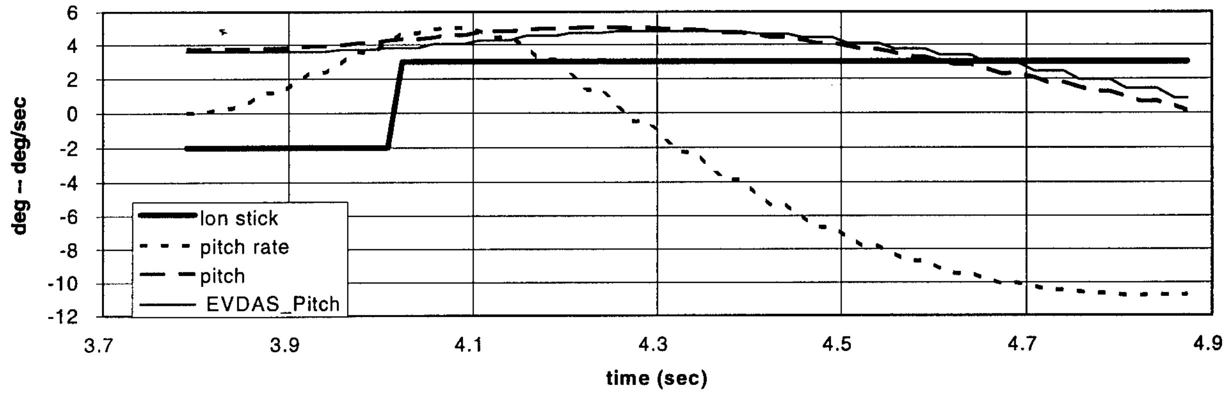
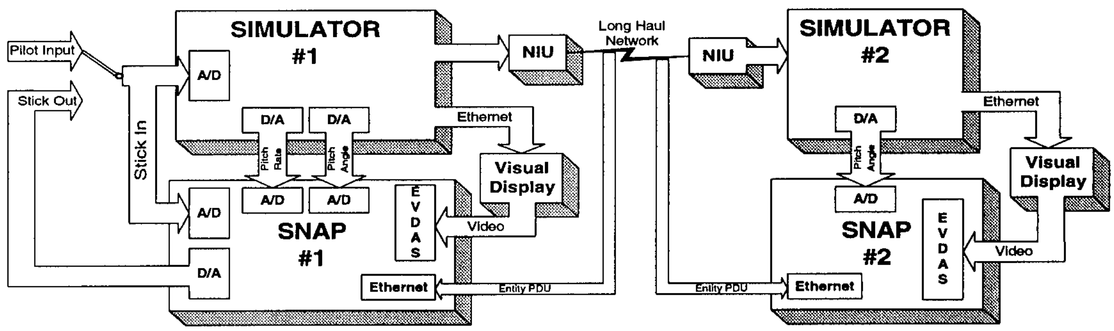


Figure 8. Single Node Pitch Response



Dual Simulator

Figure 9. Dual Node Experiment

A square wave of 0.25 Hz was applied to the stick input at simulator 1. Figure 9 shows the results of this testing. This graph shows the transport delay between the true pitch angle and the visual pitch angle at simulator 1 to be about 50 ms. This graph also clearly shows the effect of network delay and the actions of the dead reckoning algorithm on the displayed pitch angle at simulator 2. The delay between the perceived pitch angle at simulator 1 and the perceived pitch angle at simulator 2 is about 200 ms. The effect of the dead reckoning algorithm being run at a 10Hz rate at simulator 2 is indicated by the pronounced stepping in the measured display pitch angle and the overshoots in that measurement when the true pitch angle makes a significant change in direction.

CONCLUSIONS AND FUTURE APPLICATIONS

Initial experiments have demonstrated SNAP's potential for making accurate network simulator measurements. Some basic experiments and analysis work have been made as described in this paper. SNAP can provide the human factors engineers with valuable simulation performance information which they need to asses whether a specific simulation task can or cannot be performed on a certain network. SNAP can also be used to identify areas which need improvement; total simulation latency can be accurately allocated among several subsystems.

The next experiment for SNAP will be to demonstrate 3-node measurements. Three SNAP units will be located at different network simulation nodes. Performance of three dynamic vehicles will be measured and the effects of simulation delays as a function of task determined. In the future, SNAP may be used to compare the performance of various types of simulation networks. For example, SNAP could measure the performance of different network architectures such as point-to-point, distributed, and local area networks. It could demonstrate the effects of various bandwidths and determine how the simulation fidelity is affected.

Several standard tests will be developed to quantify simulation latency and performance as SNAP experiments continue. These tests will be used to baseline individual simulator performance as well as overall interactive simulation network performance.

It is hoped that SNAP will become an important tool for continued simulator network analysis and understanding.

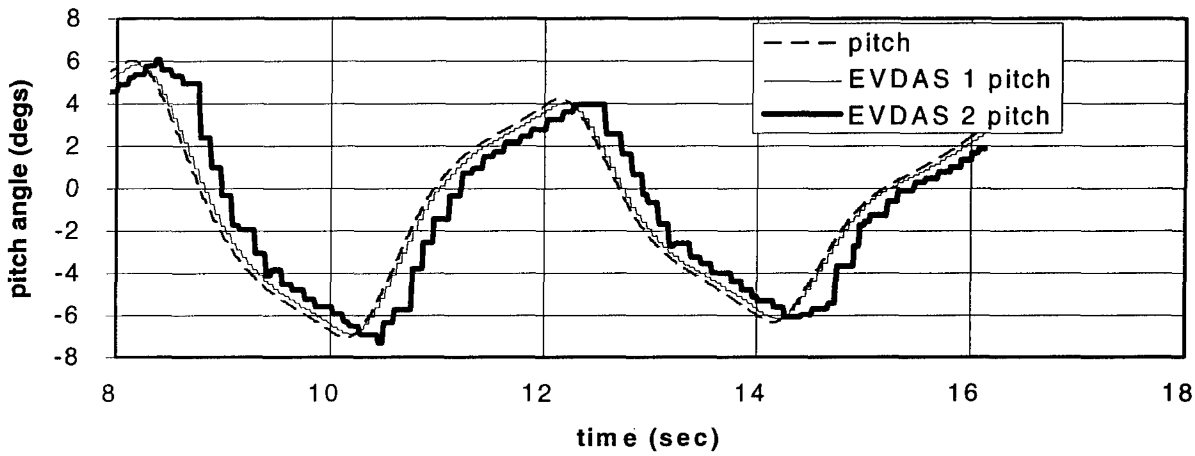


Figure 11. Dual Node Pitch Response

References

1. Slutz, G.J. and Ewart, R.B., "An Electronic Visual Display Attitude Sensor (EVDAS) for Analysis of Flight Simulator Delays", AIAA Paper No. 92-4167, August 1992.
2. Smith, R.M., "A Method for Determining Transport Delays in the Flight Simulation Environment", AIAA Paper No. 91-2964-CP, August 1991.
3. McMillan, G.R., "Cue Integration and Synchronization", January 1990.
4. Howe, R.M., "Some Methods for Reducing Time Delays in Flight Simulators", AIAA 1990.
5. Johnson, W.V. and Middendorf, M.S., "Simulator Transport Delay Measurement Using Steady-State Techniques", AIAA Paper No. 88-4619, September 1988.
6. Gum, D.R. and Martin, E.A., "The Flight Simulator Time Delay Problem", AIAA Paper No. 87-2369, August 1987.
7. Ewart, R.B., "Time Delay Measurements for Flight Simulators", SCSC Paper, July 1977.
8. Gum, D.R. and Albery, W.A., "Time-Delay Problems Encountered in Integrating the Advanced Simulator for Undergraduate Pilot Training", J. Aircraft, 1977.

Dynamic Multicast on Asynchronous Transfer Mode for Distributed Interactive Simulation

Thomas L. Gehl
Sprint, Government Systems Division
Herndon, VA

ABSTRACT

The concept of Distributed Interactive Simulation (DIS) needs advanced network technologies and services to communicate real-time state updates between autonomous simulators. The network architecture, consisting of these technologies and services, must provide high throughput messaging between multiple peer simulators that create the virtual environment. As the number of entities in the virtual environment increases, the message throughput becomes a major performance issue. Recently, scalability estimates and analysis have been performed as to how to handle tens of thousands (up to 100,000) of entities in a distributed interactive simulation scenario. Filtering techniques have been studied to determine how the message interaction between the distributed simulators can be reduced. These filtering techniques need to be performed, as appropriate, with commercially available network services to ease interoperability and enable migration to future technologies.

This paper discusses an architecture that incorporates dynamic multicast over Asynchronous Transfer Mode (ATM) to reduce the state update traffic between the distributed simulators. In discussing a dynamic multicast approach, the DIS multicast requirements of multipoint communication, group addressing, group definition, group membership, and group change rate are defined. These requirements are then applied to a network architecture consisting of a baseline topology and functional capabilities. Finally, the method of scaling DIS applications up to 100,000 interactive entities through the integration of the proposed network technologies and services is presented.

BIOGRAPHY

Mr. Gehl works on advanced networking projects for Sprint's Government Systems Division in Herndon, VA, as a Systems Design Engineer. He has actively participated, since August, 1989, on the Standards for the Interoperability of Defense Simulators, and was involved in the Communication Architecture and Security subgroup since its inception. Mr. Gehl was IBM, Federal Systems Company's lead network systems engineer on the winning Close Combat Tactical Trainer proposal, which involved designing a multicast capability over a FDDI network architecture. He has performed research on designing dynamic multicast over Asynchronous Transfer Mode technologies for the applications of Distributed Interactive Simulation.

Mr. Gehl has published and presented five papers at conferences concerning network requirements, packetized voice, and network performance evaluation with respect to DIS applications. His most recent publication, "Mapping Distributed Interactive Simulation Network Requirements onto Broadband Networks and Technologies, was presented at the SuperCom/ICC '94 conference. Mr. Gehl received his Bachelor of Science degree in Electrical Engineering and a Master of Science degree in Systems Engineering from Virginia Polytechnic Institute and State University (i.e. Virginia Tech).

Dynamic Multicast over Asynchronous Transfer Mode for Distributed Interactive Simulation

Thomas L. Gehl
Sprint, Government Systems Division
Herndon, VA

DIS ARCHITECTURAL CONCEPTS

"Distributed Interactive Simulation (DIS) is a time and space coherent synthetic representation of world environments designed for linking the interactive, free play activities of people in operational exercises. The synthetic environment is created through real-time exchange of data units between distributed, computationally autonomous simulation applications in the form of simulation, simulators, and instrumented equipment interconnected through standard computer communicative services. The computational simulation entities may be present in one location or may be distributed geographically." ("Standard for Distributed Interactive Simulation - Application Protocols," IEEE 1278, p. 1, version 2.0, fourth draft, Feb. 4, 1994)

DIS is an architectural concept based on autonomous simulations interacting in real-time between locally and widely distributed hosts. The communication services required to support DIS are defined from the following architectural concepts:

- No central computer to control the entire simulation exercise
- Autonomous simulation applications are responsible for maintaining the state of one or more simulation entities
- A standard protocol, Institute for Electrical and Electronics Engineering (IEEE) 1278, is used for communicating "ground truth" data
- Perception of events or other entities is determined by the receiving applications
- Dead reckoning (i.e., extrapolation of position and orientation in time) algorithms (DRA) are used to reduce communication processing.

DIS COMMUNICATION SERVICES

The communication service to support the DIS architectural concepts has traditionally been connectionless messaging between peer simulators. The DRAs used to provide application reliability allows for a connectionless messaging service to deliver real-time data between autonomous applications. The DIS compliant applications provide the reliability of information exchange, where needed; thus, the communication does not require a reliable transport service to ensure the successful reception of the packets. Also, the physical network technologies continue to greatly reduce the probability of error over the medium, resulting in infrequent loss of information.

The autonomous simulation application exchanges real-time state updates between peer entities. The real-time interaction of autonomous applications requires that each simulation application simultaneously communicates its state change to every other simulation entity. The state update information (i.e. Entity State PDU) must be broadcasted onto the network to simultaneously communicate between all simulaton entities. The broadcast mechanism enables the simulation applications to perform the network call only once to communicate its state updates to all of the entities that it may affect. The broadcast mechanism requires significantly less in-host

processing to communicate state updates.
Typically, state updates need be communicated to a specific group of entities based on the state information and the respective simulation states. The simulation applications must discard broadcasted state information that are received from the network and not used at the application. Receiving unneeded DIS Protocol Data Units (PDUs) requires host processing that can cause throughput bottlenecks and additional tail circuit bandwidth, that adds cost to the system life-cycle. The DIS systems need a communication service which enables the applications to communicate to specific groups, thereby reducing the host processing and tail circuit bandwidth.

The group addressing is used by the receiving entity to determine what information it must receive, and by the ATM network architecture to determine which virtual circuits must be established to communicate to the appropriate hosts. New DIS entities may join and leave its group association based on its state. The network service that allows a host to dynamically join and leave group address membership based on its application states is called dynamic multicast. The network architecture needs to provide a dynamic multicast service, which supports the functional and performance requirements of large scale advanced distributed simulation.

The communication services to support these DIS requirements include the following:

- Data transfer between each simulation application occurs in a single operation
- Unicast, broadcast, and multicast delivery of packets
- Best effort service with rare occurrence of failures (e.g., no reliability provided by the transport protocol)
- Packet integrity by detecting transmission errors associated with the network and not delivering corrupted packets to the simulation applications
- Throughput and delay performance requirements to minimize network delay and delay variance
- Flexible inclusion and exclusion from the network in a dynamic manner based on applications.

DIS SCALABILITY

The throughput and bandwidth requirements to support the communication of state updates increases dramatically as the number of simulation entities increase in an exercises. Today, there are requirements to support over 1,000 entities and to grow up to 100,000 entities.

Using the Advance Research Project Agency's (ARPA) Simulation Network (SIMNET), experiments have been performed for land and air vehicles on the entity state update rates in particular mission scenarios (Gehl, Thomas L. et al, "Interdependence of Training Utility and Network Performance using the Armstrong Laboratory Multiship Research and Development System", 15th Interservice/Industry Training Systems and Education Conference (I/ITSEC), p. 640, Dec. 2, 1993). The PDU used to update the entity state information for SIMNET is called the Appearance PDU. The average update rate for the Appearance PDU is approximately 2- to 3- times per second in engaging activities for land vehicles; and 9- to 17- times per second for air vehicles. The corresponding DIS Entity State PDU is drawn from the Appearance PDU and its updates rate can be estimated to be the same.

We estimated the bandwidth and throughput requirements for a 100,000-entity exercise integrated across 50 sites. Brigade level forces of 2,000 combat and logistics entities at each site where interconnected through a fully meshed backbone network. The 2000 entities at each site are distributed as:

- 100 manned entities at 2 PDUs/sec
- 320 computer generated forces entities at 1 PDU/sec
- 1580 support entities at 0.5 PDU/sec

The number of hosts to generate the 2,000 entities was estimated as approximately 100. The number of hosts is important in terms of managing the multicast addressing and setting-up the ATM virtual circuits. Figure 1, "ATM Connectivity for 100,000 Entity, 50 Site Scenarios," shows the ATM network connectivity of 50 sites of 100 hosts each (for simplicity, only 8 hosts are shown per site). We calculated the bandwidth and throughput requirements when entities broadcast their state updates at the given rates for a proposed Desert Storm scenario (FM 100-5 Operations, p. 6-18, June 14, 1993). The 100,000 broadcasting entities require a packet throughput of 65,600 packets per second (pps) and a network bandwidth of 134 Mbps for 256 byte packets. The interrupt processing to handle 65,600 pps at the host is very demanding, and the tail circuit costs for 134 Mbps links can become expensive. Thus, a network service that reduces the host interrupt processing and tail circuit bandwidth is needed for a large scale DIS system.

Geographical visual-detection-range filtering can reduce these performance requirements. The scenario database can be divided into 3.5 meter grids (i.e., line-of-sight distance) that allows the pre-defined domains to be mapped to multicast group addresses. These group address memberships are established at the local host and managed across the network architecture. These defintions can be performed for other states such as radio channels, exercise identification, and command and control information. As an entity enters a grid, it joins itself to that grid's corresponding group address plus the surrounding grid group addresses to receive all possible state updates within its line of sight. With the force distribution as given for our Desert Storm scenario, we can demonstrate significant savings on bandwidth and throughput performance requirements.

Dan Van Hooke's presentation at the 15th I/ITSEC (Hooke, Dan Van, "Scalability Tools, Technique, and the DIS Architecture," 15th I/ITSEC, p. 839, Dec. 2 1993) explained that a 90% reduction in traffic between sites can be obtained by filtering on geographical visual detection range grids.

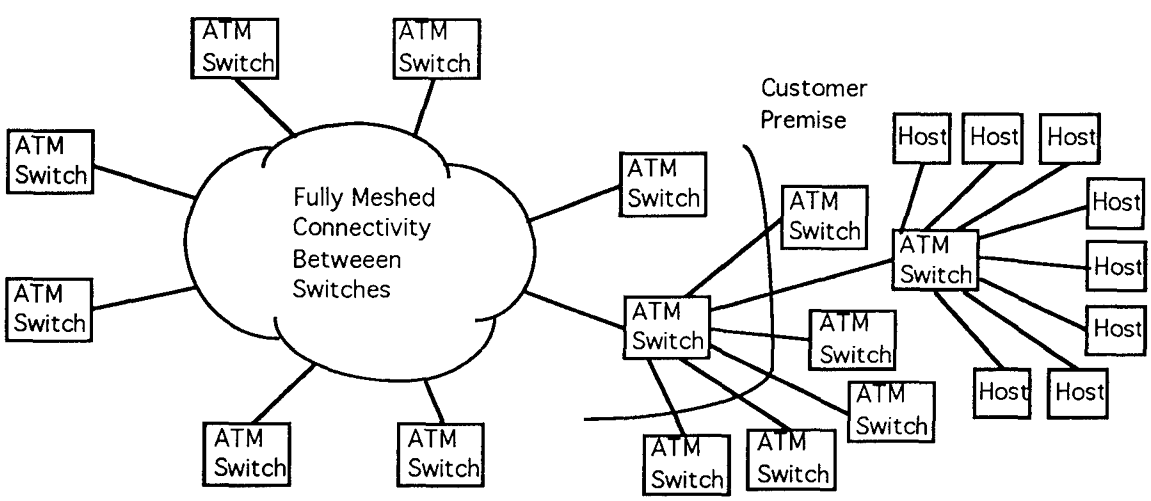


Figure 1: ATM Connectivity for a 100,000-Entity, 50-Site Scenario

This filtering results in a tail circuit bandwidth of 13.4 Mbps at 6,560 pps. For a commercial ATM network service, such as Sprint's, the user only pays for the access rate plus a usage rate to the network service and does not pay for the backbone bandwidth across the distances between the sites. Thus, the tail-circuit traffic reduction significantly reduces the life-cycle costs of a DIS system. The integration of broadband technologies (i.e., ATM and SONET) will effectively support the DIS performance requirements across a commercially available ATM multicast service. Today, Sprint provides a 45 Mbps (i.e., DS-3) ATM port access between sites and is implementing a 2.4 Gbps (i.e., OC-48) Synchronous Optical Network backbone between its ATM switches.

Figure 2, " Visual Detection Range Groups," shows a database section divided into 3.5 km grids for visual detection filtering. Entity A joins the group address of the grid it is in plus the addresses of the adjacent grids. When Entity A is in grid or group 8, it becomes a member of groups 1, 2, 3, 7, 8, 9, 13, 14 and 15. As Entity A moves from group 8 to group 9, it must join groups 4, 10 and 16, and leave groups 1, 7 and 13. The application simulating Entity A will only receive DIS PDUs communicated to groups that it is a member; thus, filtering the remaining PDUs communicated to the other group addresses.

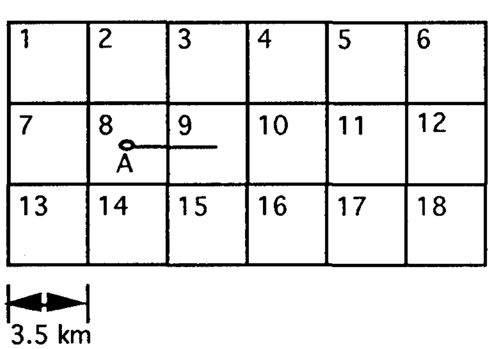


Figure 2: Visual Detection Range Groups

As we go further into the network architecture, we can filter even more based on the interaction of adjacent Brigades simulated by the hosts. Analyzing the overlap of forces in our Desert Storm scenario, we estimated a 30% reduction in traffic to the host machines using 3.5 km geographical groupings. This reduces the network bandwidth to the hosts to about 4.0 Mbps and the host throughput to about 1,965 pps, which can be supported with today's technologies. Our calculations of reducing traffic between adjacent hosts is based on the assumption that when laying the brigades out across a front only 30% of adjacent brigades appear in the others' visual detection grids.

Although the overall traffic of 100,000 entities interacting in real-time is approximately 134 Mbps at 65,600 pps, using a simple multicast function on assigned geographical locations for a DIS exercise can reduce tail circuit bandwidth to 13.4 Mbps at 6,560 pps, and host throughput to 1965 pps at 4.0 Mbps.

DIS MULTICAST REQUIREMENTS

Commercial multicast services can be used to significantly filter DIS state update traffic required to implement a large-scale advanced distributed simulation system. Figure 1 portrays a typical ATM connectivity for a particular multicast scenario. Multicast requirements must be defined to design a dynamic multicast architecture to support DIS scenarios.

The DIS architectural concepts require that all entities communicate to all other entities; or at least to many entities, based on DIS states. The many to many communications requires a multipoint-to-multipoint (N to N) as opposed to a point-to-multipoint (1 to N) multicast service. Today, the point-to-multipoint multicast service is most often discussed in support of applications such as video on demand, where a central server provides multipoint delivery of information to multiple hosts. The DIS architecture does not use a central server to communicate the state updates; thus, each autonomous simulation application must communicate in a multipoint manner, requiring multipoint-to-multipoint communications.

The DIS application provides reliability for the state updates; the multicast service can therefore be provided unreliably. As each individual simulation host communicates to the other groups of simulation hosts, the network can perform the state update delivery with a best effort service with rare occurrence of failures. Using fiber optic technologies to provide the backbone on which an ATM network architecture is integrated, today's networks are able to provide very rare occurrence of failure to support the best effort service for DIS.

Each entity relates the state information, such as geographical/positional information, to a multicast group address to implement efficient filtering. The group address can be a network address, such as an Internet Protocol (IP) multicast address, that can be mapped to virtual circuits of the ATM network. The possible domains in which an entity can exist when using geographical groupings can be pre-defined by dividing the database on which the scenario will take place at exercise set-up. Thus, the geographical domains can be pre-defined and the applications can map the states they enter to the multicast group address. This group identification of addresses is classified as determinate group addressing versus indeterminate group addressing. For indeterminate group addressing, the applications must negotiate between each other and create real-time definition of the addresses across the network architecture This creation of group addressing can become very complex as the number of real-time entities increases on the network architecture; vying for the definition of a sequence of address bits with the other simulation applications. Determinate group addressing is a very important requirement for the feasibility of implementing a multicast network service for DIS.

As previously discussed for DIS scalability, the group addressing is used by the receiving entity to determine what information it must receive, and by the ATM network architecture to determine which virtual circuits must be established to communicate to the appropriate destination hosts. This aproach reduces the host interrupt processing and the tail circuit bandwidths. An application may want to communicate to other state groupings of which it is not a member. An example of this for DIS is when an entitiy wants to fire and control a guided missile that goes into a geographical area beyond its visual detection range - to a group of entities of which it is not a member. The firing entity will want to send state updates to a group of entities even though it is not a member of that group for receiving state updates. This multicast requirement that the host be able to communicate with groups which it is not a member is called open versus closed groups.

The real-time state changes in the DIS scenarios require the network architecture to support a dynamic versus a static multicast service. Entities constantly move in and out of states in real-time. The network architecture must allow the entities to join and leave a group address in real-time based on their current state. This real-time membership of group addresses is called dynamic multicast. The dynamic multicast requirement is a driving requirement in terms of functional and performance implementation of the network services with respect to the ATM virtual circuits.

There are also multicast performance requirements in terms of managing the membership updates across the network architecture in a timely manner. The

DIS community needs to perform further evaluation to determine the group change rates for large-scale scenarios, that affect the method of updating the network switches. The total number of possible groups per exercise (i.e., the determinate groups at exercise set up) drives the number of bits needed in the network group address; and the number of active groups at the switches, access nodes, and hosts determines the amount of address memory needed on the network interface cards and switches to support an interactive exercise.

The following multicast requirements have been defined for DIS exercises:

- Multipoint-to-multipoint communications
- Unreliable delivery
- Determinate group definition
- Open multicast groups
- Dynamic multicast membership

These requirements will be mapped to the ATM technologies and architecture proposed to support large-scale distributed simulation.

DYNAMIC MULTICAST OVER ATM

The dynamic multicast service at the hosts must allow the application to join a network address that is used to receive the PDUs at the network interface card. For local area networks (LANs) such as Fiber Distributed Data Interface (FDDI), we have defined an example of how application state information can be mapped to the LANs Media Access Control (MAC) group address and joined or deleted from the memory space of the network interface card. See Figure 3, "FDDI Group Addressing." Because the application receives information based on the MAC group addresses, the network interface card can filter in hardware the unneeded PDU; greatly reducing the interrupt processing required to receive the needed PDUs.

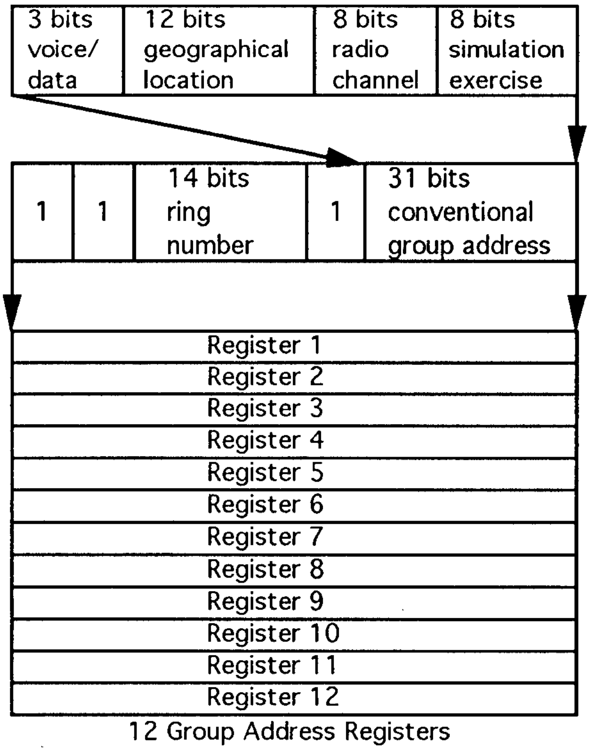


Figure 3: FDDI Group Addressing

As shown in Figure 3, "FDDI Group Addressing," the 48-bit MAC address has 31 bits available to support a sequence corresponding to application information. We have laid out the following fields that could possibly be used to filter application state information: exercises, radio channel, location, and voice or data. By defining the state meaning of those fields, an application can join the appropriate group addresses that describe its state in the exercise. Thus, the application will only receive information with respect to its states. In this example, the number of simultaneous states of which an application could be a member is limited to the 12 group address registers provided by the memory space of the adapter card.

Like FDDI, ATM has facilities that enable it to implement multicast service very efficiently. First, due to its cell implementation, each ATM cell can be switched across the network architecture through different paths to different destinations. The cell switching enables ATM technology to support the integration of multiple virtual circuits onto a backbone of physical circuits. Within the ATM cell header, the Virtual Path Identifier (VPI) and the Virtual Channel Identifier (VCI) enable the identification of the logical channels which the cell will take. The vitual path routes the cell through the network between the customer premise and network provider equipment. The virtual connection establishes an end-to-end connection on which the user can send data. In the multicast implementation on ATM, the VPI/VCI will be used to switch virtual circuits, in real-time, to provide the connectivity between the distributed applications.

To integrate the applications across existing IEEE 802.2 interoperable LANs to the ATM technology, a network service that supports both the communication and manageability of the multicast messages will be needed. Today, IP multicast exists for sending the muticast messages; and protocols such as Multicast Open Shortest Path First (M-OSPF) are being defined and tested to support the maintanance of the group membership. Using IP multicast protocols as the network service requires the state information to be mapped to the number of bits that the next generation of IP protocol provides for the group addresses. Today, IP multicast provides 16 group address bits.

The IP multicast group address must then be translated to the 8 VPI fields and 16 VCI fields to establish the virtual circuits through the ATM network. The multicast implementation between IP protocols and ATM virtual circuits is still being research. Figure 4, "ATM Group Addressing," shows how a message can be switched through an ATM virtual circuit based on a specified VPI/VCI with respect to a IP multicast group address. An incoming message is switched in hardware to the multicast virtual connetions as denoted by the VPI/VCI field.

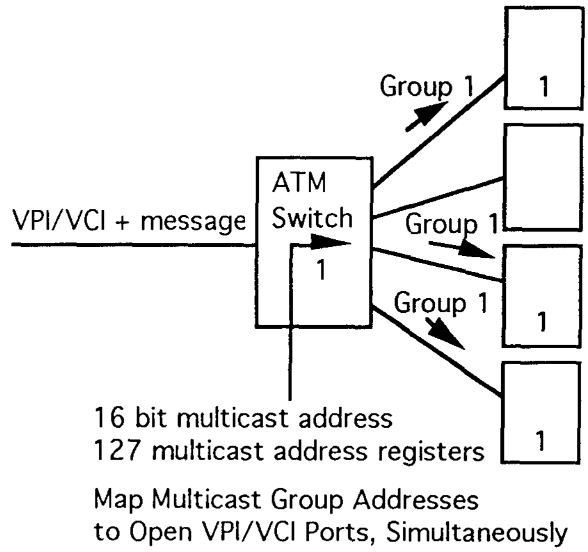


Figure 4: ATM Group Addressing

In implementing an end-to-end multicast service across the ATM network architecture, we must define an architectural approach(s) for mapping the IP multicast group addresses, as "called" from the simulation applications, to the ATM switched virtual circuits (SVCs). The approach can be based on managing the virtual circuits at each host or managing the virtual circuits from a multicast manager at a "virtual site." In our approach, we chose to define virtual sites to cluster DIS host applications onto a physical ATM termination point from the service cloud, from which ATM SVCs can be set-up. The virtual sites enable better manageability and efficiency in setting up the virtual circuits. By using the multicast manager, we have the flexibility of supporting existing applications on IEEE 802.2- compliant LANs, and we can manage simultaneous connections based on the application states.

Note that this architectural approach, like others requires additional research with respect to the functional and performance requirements defined for DIS multicasting.

For each virtual site, the multicast manager terminates the virtual circuits for a set of DIS applications. As the distributed hosts need to communicate between virtual sites, the multicast manager will use the IP address to determine the ATM destinations (i.e., termination points) and request the necessary connections to the corresponding ATM switch or its connection manager. The IP multicast manager will keep track of its local members of a particular group address and communicate to the other multicast managers its existing group address memberships.

The multicast manager sends the IP packet to the destination multicast manager as determined from the dynamically updated multicast address tables, and requests virtual circuit(s) to be established. The ATM switch or its connection manager knows the VPI/VCI destination of each multicast manager and establishes a virtual connection through the ATM switches. The receiving multicast manager sends the DIS PDU to its local hosts on the LAN (i.e., could be an ATM LAN) based on the IP group address of the message The hosts receive only the IP group addresses that match the group addresses of which they are members. See Figure 5, below.

Once the virtual circuit is established, the ATM switches simultaneously communicate to the specified circuits. Figure 4, "ATM Group Addressing" shows multicast cells switched in hardware, reducing the latency and tail circuit bandwidth of the WAN. Sprint's commercial ATM network service will charge based on the tail circuit access rates to the ATM network, rather than the backbone bandwidth created. At the local sites, the hosts only receive messages communicated to them by the multicast manager; thus, reducing the throughput processing of receiving the state updates.

The signaling between the multicast manager and ATM switches can be either out-of-band or in-band to set up the connections. Out-of-band signaling is used today in the connection-oriented networks of voice communications. In our architectural approach, the multicast manager will send a connection request message to the ATM connection manager as it determines the destination group address of the incoming IP packet. For in-band signaling, the multicast manager will either use the information within the IP multicast packet fields to communicate the connection set up to the ATM switch or encapsulates the IP multicast message into a PDU that includes the signaling.

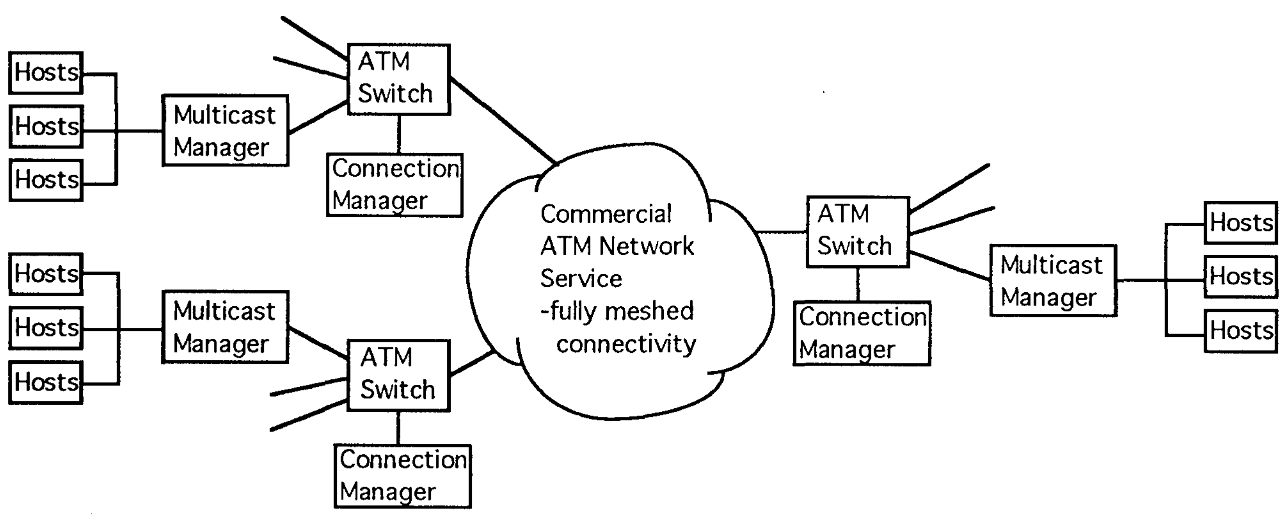


Figure 5: Dynamic Multicast Implementation over an ATM Network

This could cause complexity in the development of the ATM switches to support this SVC functionality. In either case, the SVC connection set-ups must occur within the performance requirements of the DIS communication service. Thus, more research needs to be performed to determine the best multicast mechanisms with respect to the standards being developed for SVC signaling.

SUMMARY

By relying on the functionality and performance of commercial ATM services, the DIS community will be able to more efficiently and cost effectively implement dynamic multicast on WANs to support large scale advance distributed simulation. As the functionality of ATM networks are enhanced, DIS systems will be able to migrate from today's private networks to commercial ATM services. This migration will significantly save the distance and backbone bandwidth costs in implementing today's private networks. A commercially available ATM network multicast service will accelerate, due to cost savings and increased performance, the integration of DIS applications across WANs.

The DIS multicast requirements have been defined as multipoint communications, unreliable delivery determinate group definition, open multicast groups, and dynamic multicast membership. Further research needs to be performed on multicast performance requirements to design network implementations across broadband technologies. A commercial network service, implemented to satisfy these requirements, will provide an ubiquitous WAN to integrate DIS applications.

ACKNOWLEDGEMENTS

I want to thank my management for supporting my effort in producing this paper. I also want to thank Joe Brann, my "bird dog," for doing such a thorough job in reviewing my draft paper.

THE IRIS ARCHITECTURE: INTEGRATING CONSTRUCTIVE, LIVE, AND VIRTUAL SIMULATIONS

Jason Paul Kazarian
Marjorie Anne Shultz
Naval Air Warfare Center
China Lake, CA

ABSTRACT

The Internetted Range Interactive Simulations (IRIS) project links three functionally and geographically disparate simulations via the Distributed Interactive Simulation (DIS) protocols. This paper describes the IRIS system architecture, including components portable to other projects. The paper also raises issues regarding integrating simulations into DIS and discusses candidate solutions to those issues.

IRIS integrates three simulation nodes: constructive, live, and virtual. These nodes were developed independently, without regard towards interoperability. The architecture allows all three nodes to participate in a joint DIS exercise, while minimizing modifications to the nodes themselves.

Each of the three nodes has unique characteristics. For example, the constructive node executes war games, the live node receives multiple data streams, and the virtual node incorporates avionics hardware. The paper discusses these characteristics for IRIS simulation nodes in particular. Similar projects may find these characteristics apply to other nodes in general.

Issues raised during integration are discussed. These include model, interface, and operator knowledge fidelity for all three nodes. Issues unique to each simulation class are also addressed. Further, some lessons learned are presented for the benefit of others attempting similar projects.

For further information on the IRIS project, please contact Clifford H. Stone, C0243, Naval Air Warfare Center, 1 Administration Circle, China Lake, CA 93555-6001, telephone (619) 939-2353.

ABOUT THE AUTHORS

Jason Paul Kazarian is a Computer Scientist at the Naval Air Warfare Center. He is the Software Engineer responsible for developing the IRIS Simulation Interface Unit Portable Component. Mr. Kazarian holds a BS degree in Computer Science from the California State University. He specializes in the use of object oriented software analysis, design, and implementation methods as applied to simulation systems.

Marjorie Anne Shultz is a Computer Scientist at the Naval Air Warfare Center. She is the Software Engineer responsible for developing the IRIS Simulation Interface Unit for the Weapons and Tactics Analysis Center constructive simulation. Mrs. Shultz holds a MS degree in Audiology from Purdue University and a MS degree in Computer Science from the California State University. She has extensive war gaming analysis and object oriented programming experience.

THE IRIS ARCHITECTURE: INTEGRATING CONSTRUCTIVE, LIVE, AND VIRTUAL SIMULATIONS

Jason Paul Kazarian
Marjorie Anne Shultz
Naval Air Warfare Center
China Lake, CA

INTRODUCTION

The Internetted Range Interactive Simulations (IRIS) project links three functionally and geographically disparate simulations via the Distributed Interactive Simulation (DIS) protocols. The project, scheduled for completion in December 1994, is supported by the Defense Modeling and Simulation Office and three sites of the Naval Air Warfare Center: China Lake, CA; Orlando, FL; and Point Mugu, CA.

One project goal is to provide a set of portable architecture components enabling others to adapt existing simulation nodes into DIS exercises. This paper, presented for the Sixteenth Interservice-Industry Training Systems and Education Conference, describes the IRIS system architecture and its reusable components. The paper also discusses issues and lessons learned in reference to integrating simulations with DIS, as well as candidate solutions to those issues.

IRIS SYSTEM ARCHITECTURE

IRIS integrates simulation nodes originally developed without regard towards participating in a distributed environment. The IRIS architecture provides hardware and software to conduct a DIS exercise while minimizing modifications to the simulation nodes themselves.

The architecture core consists of a simulation node at one end and a Network Interface Unit (NIU) at the other (Figure 1). Between these elements is a Simulation Interface Unit (SIU), which consists of two software components: one portable, one simulation specific. Three of these cores communicate via the Network Infrastructure to form the IRIS system. This architecture is described in the following paragraphs.

Simulation Nodes

IRIS integrates three simulation nodes: a constructive war game simulation, a live air and sea test range, and a virtual avionics integration facility. Each of these nodes participates equally in IRIS exercises, allowing (for example) a live aircraft to engage a virtual aircraft with weapon dynamics being modeled constructively.

Constructive Node. IRIS integrates the Weapons and Tactics Analysis Center (WEPTAC), located at the China Lake main site, as its constructive node. WEPTAC is a Man In The Loop (MITL), battle level war gaming simulation developed before the existence of DIS. WEPTAC employs a central computer that tracks the state of all simulation entities under its control.

WEPTAC is a stand-alone simulation that ignores most key DIS concepts.[1] IRIS provides the support necessary to enable interaction with external elements, allowing for the development and execution of more complex exercises.

Live Node. IRIS integrates the Battle Management and Interoperability Center (BMIC), located at the Point Mugu sea test range, as its live node. BMIC is a test facility providing electronic and human command and control links to air and sea assets for use in conducting tactical exercises, both with and without live weapon fire.

IRIS integrates BMIC computer assets to provide communication between BMIC and other nodes. BMIC operators view synthetic assets as if they were live, a new capability.

Virtual Node. IRIS integrates the Weapons Software Support Facility (WSSF), located at the China Lake air field, as its virtual node. WSSF is a F/A-18 aircraft test facility providing MITL opera-

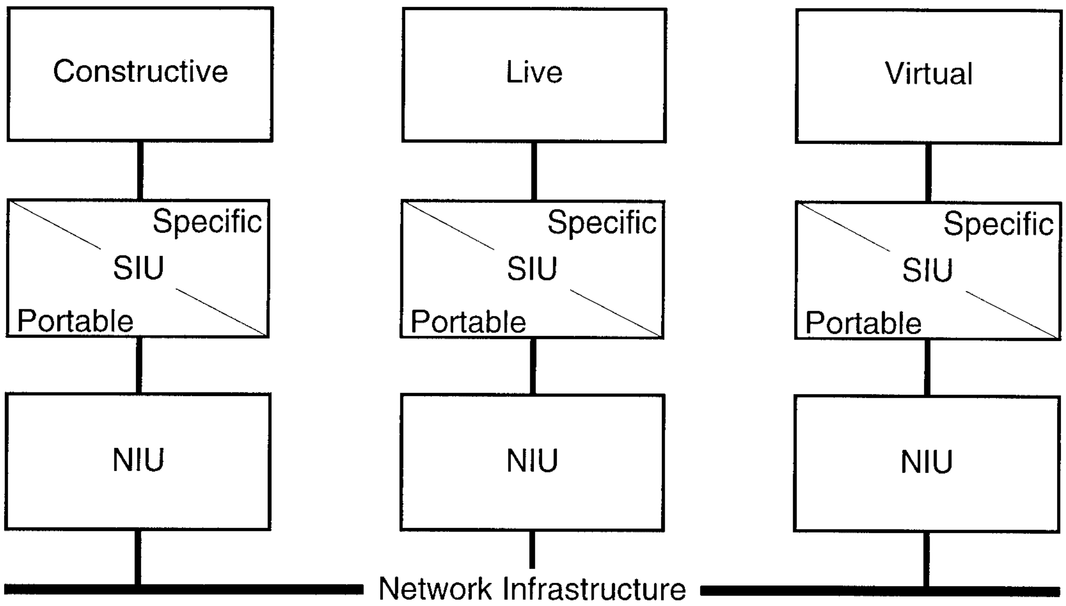


Figure 1: IRIS Architecture Block Diagram

tion. WSSF uses software models that drive F/A-18 avionics to support subsystem integration.

WSSF is currently used to test target engagement and weapon firing, but not weapon dynamics. Integrating WSSF into IRIS adds support for mission level pilot training.

Simulation Interface Unit

IRIS has a design requirement to minimize modifications to the simulation nodes. IRIS also has a product requirement of providing reusable components for integrating other nodes into DIS exercises. Consequently an architecture element that isolates the internal workings of a simulation node from the DIS protocols is needed. That element is the SIU.

The SIU incorporates two software components: portable and simulation specific. Other projects may reuse the IRIS SIU Portable Component (PC) directly. A Simulation Specific Component (SSC) will need to be custom written, perhaps using IRIS documentation as guidance.

Simulation Specific Component. The SIU SSC extracts data from a simulation node and invokes its associated PC to communicate with other IRIS simulation nodes. An SIU is tightly coupled to a given simulation, so the SSC software is not directly reusable on other simulation nodes.

The SSC design varies depending on the node design. In IRIS, for example, the BMIC and WSSF SIU elements implement their corresponding SSCs via data conversion and communication. The WEPTAC SIU element, on the other hand, is more complex: because WEPTAC has no external interface, its SSC appears to the node as if it were a human player. In general, a SSC for single entity simulation node will be relatively easy to develop compared to a SSC for a multiple entity node.

Portable Component. The SIU PC is a Protocol Data Unit (PDU) network shell implemented via an Ada package interface. This shell complies with the DIS 2.0.4 draft standard and will be modified as the specifications evolve. Four general purpose PDUs are supported: Collision, Detonation, Entity State (E/S), and Fire. The Acknowledge and IsPartOf PDUs[2] are supported for entity handover and transfer of control.

The SIU PC handles PDU storage and forwarding to and from its associated NIU. Object-oriented software supporting generic instantiation, allowing the use of abstract data types, is employed.

Network Interface Unit

IRIS, like other DIS projects, allows simulation nodes to communicate over a network, so a standard interface between nodes is a critical architectural element. This function is performed by the NIU, implemented by each simulation node.[3]

Interfaces. The NIU was developed in ANSI C by our Training Systems Division in Orlando. Driven by a SIU PC, the NIU handles transmission and reception of PDU packets over the IRIS network infrastructure. A simulation node's NIU communicates with its associated SIU PC via shared memory and the other two NIUs via the network.

Functions. The NIU provides three functions to reduce simulation node integration burdens. First, the NIU performs dead reckoning for its associated simulation node and issues E/S PDU updates only when necessary. Second, the NIU allows PDU packet filtering based on a script file. Third, the NIU performs coordinate conversion between DIS standard and other systems.

Network Infrastructure

The project developed a private wide area network to host IRIS exercises. This network links the NIUs in compliance with the Communication Architecture for DIS (CADIS) standard.[4] Hardware configuration details are documented in a MIL-STD-100 drawing package.

Bandwidth. Aggregate network data transfer speeds are limited by the communication link among all three sites, a dedicated T1 channel running at 1.544 megabits per second. Assuming an E/S PDU length of 1664 bits (including four of the possible eight articulation parameters)[5] and an efficiency of 80%, the network offers a throughput of 742 E/S PDU packets per second.

Protocols. IRIS is being developed to comply with CADIS Phase 1, Class 1, which specifies the use of the Internet Protocol suite. This suite includes the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP is a point-to-point protocol offering guaranteed delivery; UDP is a broadcast protocol.

The DIS standards specify that PDUs shall be issued using a broadcast communication service, such as UDP. IRIS, however, uses TCP, a point-to-point service: each node employs its own subnetwork and our Ethernet routers do not yet support broadcasts across subnetworks. Reprogramming the routers to support UDP broadcasts across the subnetworks is under consideration.

Reusing IRIS Architecture Components

Several IRIS architecture elements are reusable by other projects. These are the NIU software and documentation, the SIU PC software and documentation, the SIU SSC documentation (the software is available, but may not be reusable), and the network infrastructure documentation. These items are unclassified and for unlimited distribution. Interested parties should contact the project office for further details.

SIMULATION UNIQUE CHARACTERISTICS

Each of the three simulation classes integrated by IRIS (constructive, live, and virtual) has unique characteristics that must be addressed for successful integration. The following paragraphs discuss these characteristics for IRIS simulation nodes in particular. Other projects may find these characteristics apply to similar nodes in general.

Constructive Node Characteristics

Since WEPTAC is a MITL, battle level, war gaming simulation, its SIU behaves as if it were a human player. The WEPTAC SIU interfaces with the existing command and response language. This allows external IRIS entities to appear as if they were WEPTAC platforms.

In some cases, the WEPTAC SIU interacts with the war game in the same fashion as a player. For example, a player may move an entity by issuing a command or request entity status and receive a report. These interfaces are stimulated by the WEPTAC SIU.

In other cases, the WEPTAC SIU has fewer restrictions. A player, for example, cannot obtain information on aircraft that are out of sensor range, and information on unfriendly or hostile forces is limited. But the WEPTAC SIU instead has unlimited access to all war game data for generating E/S PDUs reflecting ground truth.

Live Node Characteristics

Interacting with a live simulation, the BMIC SIU incorporates unique functions and interfaces. Unlike the other nodes, BMIC entities are not synthetic. Operators use computers for coordinating entities in an exercise, not modeling the entities themselves. Consequently the BMIC SIU must accommodate more human interaction.

Because BMIC incorporates computer, personnel, and range assets, its associated SIU must obtain inputs from several disparate sources. For example position data is obtained from range instrumentation, command and control data from operators, and environment state data from sensors. The architecture must provide a mechanism for not only distributing digital data over the network, but also exercise management commands.

Virtual Node Characteristics

Of the three simulation nodes, WSSF is the least complex. It interfaces with the F/A-18 model via Ethernet using UDP at a rate of 20 Hz. Because WSSF simulates exactly one entity, processing for the WSSF SIU is minimal.

The WSSF node is limited to processing eight targets. Consequently its SIU filters incoming targets and only forwards information on the eight IRIS targets closest to the aircraft. The target inputs are a subset of the data contained in E/S records, so the target generator is driven by E/S PDUs.

WSSF has limitations that place additional burdens on the other nodes. For example WSSF does not model weapon dynamics. IRIS exercises call for a virtual simulation node to fire weapons, so that function must be handled by WEPTAC. Coordinating weapon fire in one simulation node with weapon modeling in another is one of the integration issues discussed later in the paper.

DEVELOPMENT AND INTEGRATION ISSUES

The IRIS simulation nodes were designed for independent and unique purposes. When the three nodes are integrated, fidelity differences among the simulations must be resolved. In addition, each node has its own integration issues.

Fidelity Issues

There are significant fidelity differences among the three IRIS simulation nodes. BMIC uses live units, so it has infinite fidelity. WEPTAC uses three Degree of Freedom (DOF) models updated at one Hz. WSSF uses a six DOF model running at 20 Hz. When these simulation nodes are integrated, fidelity issues related to models, interfaces, and operator knowledge result.

Models. Model fidelity differences may give one simulation an unrealistic advantage over another. For example, live aircraft have limited fuel supplies and respond differently depending upon the payload type and weight. When a live aircraft flies with weapons on board, its maneuverability and speed are limited. But WSSF continuously models an unladen configuration and a full fuel tank. Our solution to this problem is to use pilots as operators, asking them to command simulated assets within the limits of a live asset.

Interfaces. There are no modeling fidelity issues for live entities. An F/A-18 virtual simulation, for example, cannot be more accurate than a real F/A-18. Instead overall exercise fidelity is limited by the interface between live and synthetic elements. For example, fidelity is restricted by the interaction between live entity operators and synthetic simulations: communication is limited to voice and Time-Space Position Information data, the latter having a one-third Hz output.

Operator Knowledge. A pilot flying a live aircraft is limited to inputs from radio communication and visual cockpit displays. But WEPTAC and WSSF provide an operator more information than would be received during a live mission. This information must be carefully controlled to level the fidelity between simulations, otherwise live entities are placed at a disadvantage.

Constructive Simulation Issues

Integrating WEPTAC with IRIS presents unique problems. Unlike the other two nodes, WEPTAC demands advance knowledge of all entity types that will be used in an exercise. If an external entity is to interact with WEPTAC, it must also be modeled in the war game. DIS compliant outputs, such as detonation, must also be generated.

Prior Knowledge. WEPTAC is a stand-alone simulation. So if an entity participates in an exercise, its type must exist in a data base. When a new entity is added to the game, it is declared as a data base type. These types can be created after an exercise has started, but this demands pausing the game. IRIS exercises do not include planned pauses, so the data base must include all potential entity types and weapon attachments.

Simultaneous Modeling. WEPTAC by itself does not interact with external entities, so all exercise entities must be modeled in the game. IRIS uses "remote platforms" to differentiate between constructive entities and those created by other nodes. For external entities, the owning simulation maneuvers its constructive image by issuing E/S PDUs, which the WEPTAC SIU then interprets to issue war game commands.

WEPTAC maneuvers both internal and external entities, as the simulation cannot tell one from the other. Consequently external entities appear to move in "skips and jumps," since the SIU sends updates at a one Hz interval.

DIS Compliance. Weapon detonation, which is required for DIS exercises, is not a WEPTAC output. The game instead calculates hit or miss with a Probability of Hit table and damage using a Probability of Kill (PK) table, not the detonation location. But DIS protocol requires supplying a detonation location without damage assessment.

For IRIS, WEPTAC generates Detonation PDUs by filling the PK table with null values. This provides a hit or miss report without damage assessment. If the result is a hit the SIU issues a Detonation PDU, but substitutes the target location to approximate the detonation location. After this PDU is dispatched, the simulation node owning the target generates a damage assessment.

Live and Virtual Simulation Issues

Adding BMIC and WSSF to IRIS exercises also presents both obvious and latent problems. Certainly cost and safety become major issues when live assets are involved. Less obvious, perhaps, is coordinating between a digital, synthetic environment and an analog, live environment: sometimes there is no mapping from one domain to the other.

Scenario Planning. Proper scenario management and planning are mandatory for the success of an exercise incorporating live elements. Entities must be deployed in proper sequence to be on station for the simulation start. The exercise should not plan on being able to freeze. But if in practice it does, live entities are directed by voice to speed up, slow down, or idle until the simulation resumes. The exercise manager, therefore, must know the planned positions of all entities at all times. The exercise manager must also have the ability to suspend or reveal all synthetic entities if conditions warrant.

Coordination. The major benefits of integrating live simulations with constructive and virtual simulations are reduced cost and increased safety by eliminating the need for firing real weapons. This requires a mechanism for synchronizing live and synthetic engagements.

BMIC does not synchronize assets, nor does WSSF model weapon flight. So the IRIS project developed the IsPartOf PDU to transfer modeling responsibility from one simulation node to another. IsPartOf allows a node to pass control of an entity, usually a weapon, to a node capable of modeling that entity, usually a synthetic simulation.

An IsPartOf PDU is generated at platform creation for each weapon requiring external services. After IsPartOf is received, the weapon modeling simulation node (WEPTAC in IRIS's case) monitors all Fire PDUs and determines if the transferring platform has fired a weapon. If so, the modeling node assumes responsibility for that weapon, simulates the flight path, issues weapon E/S updates, and generates a Detonation PDU upon target impact.

PROJECT LESSONS LEARNED

All development projects have an element of learning associated with them; IRIS is no exception. We pass along the following lessons, hoping others will benefit from them.

Drive Requirements from the Exercise. The DIS standard is both comprehensive and evolving. It contains facilities for modeling almost all aspects of air, surface, and subsurface warfare. Few exercises, however, need to address the entire standard. Meaningful scenarios can be built from a small subset, for example the E/S, Fire, Collision, and Detonation PDUs.

We found it useful to implement a phased build approach: concentrate our development effort on those portions of the standard necessary to meet a simple exercise's requirements, then add software as required for larger scenarios. This approach kept us working on task essentials instead of being distracted by the standard's formidable detail.

Define Requirements Before Design. Not only are the DIS standards detailed, they also dictate a compliant architecture and design. But this restriction in reality affects only the back-end communication between simulation nodes, not the top-level system interaction. Most system requirements are defined by site needs, not standards compliance needs. So DIS projects still benefit from a traditional top-down analysis.

Our first attempt at software requirements analysis was conducted via a bottom-up approach: each site submitted requirements reflecting a notional component design. We had difficulty allocating our requirements to these components and decided to perform another analysis, this time lumping all requirements in one collection, without deciding which component owned them. The second allocation attempt was smoother, and several requirements were moved from one component to another.

Tailor the Development Environment. DIS projects typically build on an organization's existing resources, so more than likely pieces of in-place software engineering processes are reused. But because DIS integrates disparate systems, there is risk that a resulting development environment will not form a cohesive whole.

We used four isolated networks while developing IRIS: the IRIS net itself, the Internet, a developer's Ethernet, and an AppleTalk net. Our software was developed under Unix, but we used Apple Macintoshes for documentation and electronic mail. Two of the five software engineers relied on IBM PC compatibles for their personal computers. Needless to say significant time was spent converting work products from one platform to another. As a whole, this extra work did not break the project, but it did impede progress.

Share the Communication Wealth. Simulation integration, or any integration project for that matter, encompasses a number of distinct disciplines: business administration, computer science, engineering, mathematics, and operations research all come to mind. Such projects present unique opportunities for the participants to learn how other topics relate to their primary discipline. Frequent communication between and interaction among all team members results in both project progress and personal reward.

We distributed our work among six teams: networking, project management, scenarios, simulations, software, and systems. We communicated well within teams using technologies such as electronic mail and video teleconferencing for frequent contact across geographic distances. But we did not take advantage of these facilities to bring whole teams together. We at first had difficulty establishing boundaries between the teams, so it took some time until people were comfortable working on multiple tasks in parallel. We eventually learned how to communicate among teams, but addressing this issue earlier in the project would have been well worth the effort.

SUMMARY

The Internetted Range Interactive Simulations project integrates constructive, live, and virtual simulation nodes to conduct DIS exercises. IRIS developed an architecture with portable components; other projects may benefit from this effort.

The IRIS architecture is built around cores that consists of three elements: a simulation node, a Simulation Interface Unit, and a Network Interface Unit. The SIU consists of two components: one common to all nodes and one simulation specific. Three cores (one for each node) communicate via the IRIS network infrastructure.

IRIS integrates simulation nodes that do not have external communication interfaces, so architecture and design tradeoffs to minimize node modifications were made. In general, simulation nodes modeling many entities were more difficult to integrate than nodes modeling a few entities.

Integrating disparate simulation nodes raised general issues in coordination, fidelity, and operator knowledge. In addition, issues unique to a specific type of simulation, such as constructive, were also raised. The IRIS architecture has evolved to solve the problems generated by these issues.

ACRONYMS

BMIC	Battle Management and Interoperability Center
CADIS	Communication Architecture for Distributed Interactive Simulation.
DOF	Degree of Freedom.
DIS	Distributed Interactive Simulation.
E/S	Entity-State.
Hz	Hertz, or cycles per second.
I/O	Input-Output.
IRIS	Internetted Range Interactive Simulations.
MITL	Man in the Loop.
M/S	Meters per Second.
NIU	Network Interface Unit.
PC	Portable Component (of Simulation Interface Unit).
PDU	Protocol Data Unit.
PK	Probability of Kill.
SIU	Simulation Interface Unit.
SSC	Simulation Specific Component (of Simulation Interface Unit).
TCP	Transport Control Protocol.
UDP	Universal Datagram Protocol.
WEPTAC	Weapons and Tactics Analysis Center.
WSSF	Weapons System Support Facility.

REFERENCES

1. Institute for Simulation and Training (1993 September). *Distributed Interactive Simulation Standards Development Operational Concept 2.3* (IST-TR-93-25), Page 4. Orlando, FL: University of Central Florida (UCF).

2. Paterson, Dana A (March 1994). Joining Distributed Entities (The IsPartOf PDU). Proceedings of the *10th Workshop on Standards for the Interoperability of Defense Simulations,* Volume II, Pages 107-111. Orlando, FL: Institute for Simulation and Training, University of Central Florida.

3. Training Systems Division. *Software Requirements Specification for the DIS Network Interface Unit.* Orlando, FL: Naval Air Warfare Center.

4. Institute for Simulation and Training (1993 June). *Communication Architecture for Distributed Interactive Simulation* (IST-CR-93-15). Orlando, FL: University of Central Florida.

5. Ibid (1994 February). *Standard for Distributed Interactive Simulation—Application Protocols, Version 2.0 Fourth Draft* (IST-CR-93-40), Page 108. Orlando, FL: University of Central Florida.

INTEGRATING CONSTRUCTIVE AND VIRTUAL SIMULATIONS

Clark R. Karr and Eric Root
Institute for Simulation and Training
3280 Progress Dr., Orlando, FL 32826

ABSTRACT

"Constructive" battlefield simulations/models typically control groups of entities (e.g. the tanks in a tank company) as aggregates rather than as sets of individual simulated entities. Constructive models also are typically "time–stepped"; that is, time proceeds in discrete steps with each step covering several seconds or minutes. The position, movement, status, and composition of aggregate units are maintained for the unit as a whole and are the result of statistical analysis of the units' actions rather than the result of the actions of individual entities.

In contrast, "virtual" simulations typically represent each tank or vehicle as a distinct simulation entity and operate in "real–time". Manned virtual simulators each represent a single vehicle. The human crews in their simulators interact in a common, simulated (virtual) battlefield through the exchange of information packets on the network that connects the simulators. Additional, unmanned entities in the virtual environment are generated and controlled by Computer Generated Force (or Semi–Automated Force) computer systems.

The interoperation of time–stepped, aggregate constructive simulations with real–time, entity level virtual simulations provides benefits to both the analytic and training communities but poses several technical challenges. This project's goal has been to demonstrate the feasibility of the interoperability of constructive and virtual simulations through the integration of the Eagle constructive model and the SIMNET virtual environment. A network architecture, network protocol, and specialized interface computer systems have been developed. Solutions to space and time correlation, to movement of entities between the two environments, to the transmission of orders and commanders' intentions between environments, and to combat between environments have been developed.

ABOUT THE AUTHORS

Clark R. Karr is the Principal Investigator of the Integrated Eagle/BDS–D Project at the Institute for Simulation and Training. Mr. Karr has earned a B.S. in Biology from the University of Denver and a M.S. in Computer Science from the University of Central Florida. His research interests and publications are in the areas of artificial intelligence and simulation.

Eric D. Root is a Software Engineer in the Integreted Eagle/BDS–D Project at the Institute for Simulation and Training. Mr. Root has earned a B.S. in Computer Science and Mathematics from Missouri Western State College; he is currently a M.S. student in Computer Science at the University of Central Florida. His research interests and publications are in the area of simulation.

INTEGRATING CONSTRUCTIVE AND VIRTUAL SIMULATIONS

Clark R. Karr and Eric Root
Institute for Simulation and Training
3280 Progress Drive, Orlando FL 32826

INTRODUCTION

"Constructive" battlefield simulations (models) typically control groups of entities (e.g. the tanks in a tank company) as aggregates rather than as sets of individual simulated entities. The position, movement, status, and composition of aggregate units are maintained for the unit as a whole and is the result of statistical analysis of the units' actions rather than the result of the actions of individual entities.

In contrast, "virtual" simulations typically represent each tank or vehicle as a distinct simulation entity and allow "Man in the Loop" interaction. The SIMNET networked training system is a prototypic distributed, virtual simulation operating in real-time. Manned simulators each represent a single vehicle. The human crews in their simulators interact in a common, simulated (virtual) battlefield through the exchange of information packets on the network that connects the simulators. Additional, unmanned entities in the virtual environment are generated and controlled by Computer Generated Force (or Semi-Automated Force) computer systems.

Interoperating constructive and virtual simulations provides benefits to two simulation communities. The analytic community can obtain detailed entity level information from virtual simulations and use that information to ground constructive models on vehicle level performance. The training community benefits by being able to conduct small unit virtual exercises within the context of a larger (brigade/ division/corps), dynamic battle. Constructive models are also used in training higher level commanders and their staffs. This training can also be enriched through the interoperation of constructive and virtual simulations by supplementing aggregate statistical interaction with entity interactions.

The interoperation of time-stepped, aggregate (constructive) simulations with real-time, entity level (virtual) simulations poses several technical challenges. Among those are space and time correlation, communication of information between simulations, and interaction between entities and aggregates. The Integrated Eagle/BDS-D project's goal has been to demonstrate the feasibility of the interoperability of constructive and virtual simulations through the integration of a constructive model (Eagle) and SIMNET.

A network architecture, network protocol, system architecture, and computer software have been developed to support constructive/virtual interoperation.

This work has been reported earlier in (Karr, 1994b); background information is repeated here for clarity and completeness. This paper reports changes to the Interoperability Protocol and the installation of the system at the AirNet facility in Ft. Rucker Alabama.

BACKGROUND

Computer based battlefield simulations and models can be divided into two broad classes, "aggregate" and "entity level", based on the granularity of the entities modeled. "Aggregate" simulations control units (e.g. the tanks in a tank company) as an aggregate rather than simulating each individual entity within the unit. The position, movement, status, and composition of aggregate units are maintained for the unit as a whole and are the result of statistical analysis of the units' actions rather than the result of the actions of the individual entities. In contrast, "entity level" simulations represent each vehicle as a distinct simulation entity. The position, movement, status, and composition of units in entity level simulations is inferred from the individual entities. Computer-supported wargames and distributed interactive simulations are typically aggregate and entity level simulations respectively (Mastaglio, 1991).

Simulations and models can also be classified on the basis of their time scales. Real-time simulations operate with events occurring at the same rate as the corresponding real-world events. Non-real-time simulations operate faster or slower than real-time.

Throughout this paper, "constructive" will apply to non-real-time, aggregate simulations and "virtual" to real-time, entity level simulations.

The differences in entity granularity and time scales among simulations create difficulties in simulation interoperability. For example, in the battlefield environment, it is difficult for entities in virtual simulations

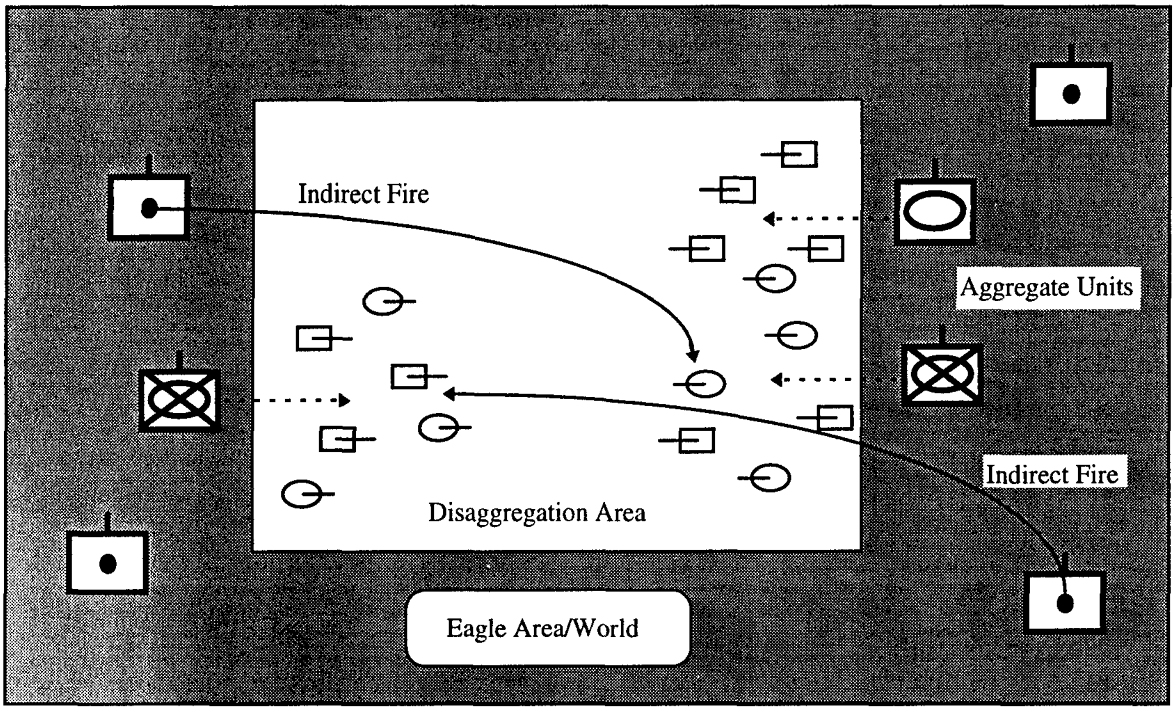


Figure 1. Typical Scenario

to detect and react to aggregate units. Similarly, units in constructive simulations typically do not detect and attack individual entities. The problems associated with differing time scales are obvious; simulations need to be operating at the same time scale for their interactions to make sense.

This paper discusses the Integrated Eagle/BDS-D project which began as a proof of concept demonstration of the interoperability of a constructive simulation (the Eagle combat model) and a virtual simulation (the Institute for Simulation and Training's Computer Generated Forces (CGF) Testbed operating in SIMNET). The project has been extended to study additional issues in interoperating constructive and virtual simulations. Three organizations are involved in this project: U.S. Army TRADOC Analysis Center (TRAC), Institute for Simulation and Training (IST), and Los Alamos National Labs (LANL). TRAC is responsible for the Eagle simulation, IST is responsible for the IST CGF Testbed, and LANL is developing the Simulation Integration Unit (SIU). Earlier work on this project has been reported in (Karr 1992a, 1993, and 1994b).

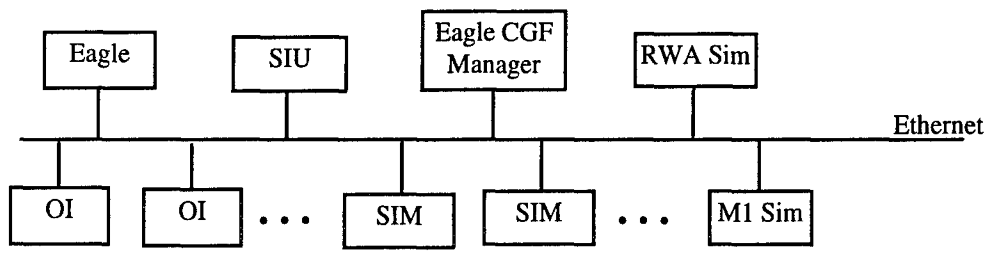
Overview of Eagle

The Eagle system is a corps/division level constructive combat model that simulates ground combat at the company and battalion level. That is, the smallest units (maneuver units) in Eagle are the company and battalion. Eagle is a combat analysis tool used for combat development studies. It is used in analyzing the effects of weapons systems, command and control, military doctrine, and organization on force effectiveness. Eagle is implemented in LISP on Symbolics and Sun workstations and runs faster than real-time. It is described in (TRAC1-7, 1993).

Overview of SIMNET/DIS

The U.S. Army/DARPA SIMNET is a well-known distributed, interactive, virtual simulation system used to train tank and vehicle crews in cooperative team tactics. In SIMNET, individual vehicle simulators are connected via a computer network, permitting them to coexist in a common, shared simulation environment and to interact (e.g. engage in combat) through the exchange of information packets on the network. SIMNET simulators usually each represent a single tank or vehicle. The documentation of SIMNET is extensive; (Thorpe,1987) and (Pope,1991) are good examples.

The Distributed Interactive Simulation (DIS) protocol is intended to replace the SIMNET protocol. The development of the DIS protocol is a cooperative effort involving the Department of Defense, industry, and academia and is being done through a series of workshops. The latest version of the DIS protocol is documented in (IST-CR-93-15, 1993).



Eagle - TRAC Eagle Model
SIU - Simulation Interface Unit
Eagle CGF Manager - IST Eagle CGF Manager
OI - IST CGF Testbed Operator Interface
SIM - IST CGF Testbed Simulator
M1 Sim - M1 Manned Simulator
RWA Sim - Rotary Wing Aircraft Manned Simulator

Figure 2. Network Configuration

In a virtual exercise, the opponent for the trainees can be provided in different ways. One technique is to provide a computer based simulation that generates and controls one or more simulation entities (e.g. tanks and infantry). Such a computer generated opposing force is usually referred to as a Computer Generated Force (CGF) or Semi-Automated Force (SAFOR).

Overview of IST CGF Testbed

IST has been conducting research in the area of CGF systems and has developed a SIMNET and DIS compatible CGF Testbed that connects to a SIMNET or DIS network and provides a mechanism for testing CGF control algorithms. The IST CGF Testbed is described in (Smith, 1992) and (Karr, 1992b). A single Testbed system consists of a pair of IBM PC-compatible computers: the Operator Interface (OI) is the human operator console and the Simulator controls the behaviors of the simulated entities requested by the human operator. OIs and Simulators communicate with one another using non-SIMNET/DIS protocol data units (PDUs) and with the other simulators using SIMNET/DIS PDUs.

INTEROPERATION OF SIMULATIONS

To demonstrate the interoperability of Eagle and CGF systems, test scenarios have been devised (see figure 1). Typically, an Eagle exercise consists of two brigade size forces with company and battalion maneuver units. Initially, no entities are controlled by the CGF system. The Eagle exercise occurs in a large area (100s km by 100s km) called the "Eagle area/world". The CGF system operates with a representation of an area of terrain called the "virtual area/world". The virtual area is within the Eagle area. The Eagle model preregisters both the virtual area and one or more "disaggregation areas" within the virtual area.

When an Eagle unit enters a disaggregation area, the unit is "disaggregated" into its component entities. The component entities are instantiated as virtual entities under control of CGF systems and the Eagle unit becomes a "shadow" or "ghost" unit (maintained but not controlled by Eagle). Combat occurs in the virtual area between virtual entities with the participation of indirect fire from Eagle units outside the virtual area. When a disaggregated unit moves outside its disaggregation area, the unit is "reaggregated"; i.e. the virtual entities are removed and Eagle assumes complete control of the unit.

Responsibilities

During the course of the scenario, the components of the system have distinct responsibilities. The Eagle model simulates the constructive world, determines when units are to be disaggregated and reaggregated, sends/receives information to/ from the virtual world, and, while any unit is disaggregated, shifts to real-time execution.

The CGF systems are responsible for responding to disaggregation/reaggregation requests, for simulating individual entities within the virtual world, and for sending/ receiving information to/from the constructive world.

The CGF operator is responsible for initiating, monitoring, and controlling simulated entities' behavior in the virtual world, for following the orders of the Eagle model, and for informing the Eagle model of his/her unit's progress in the battle.

The SIU coordinates the communication between Eagle and the CGF systems and maintains summaries of the disaggregated units' location and composition.

Network Configuration

Figure 2 shows the network configuration. The Eagle model, the SIU, the CGF Operator Interfaces (OIs), and the CGF Simulators are nodes on the network. All communication between Eagle and the CGF systems is mediated by the SIU as described below. Eagle and the SIU communicate with via Remote Procedure Calls (RPCs). The SIU and the CGF nodes communicate with using an interoperability protocol (described below). Because the SIU translates, stores and forwards information between Eagle and the virtual world, this paper will discuss primarily the SIU and CGF protocol with the understanding that the information is then relayed between the SIU and Eagle in via RPCs.

The SIU, OIs, and Simulators also receive virtual world PDUs from all nodes on the network. The OIs and Simulators process virtual world PDUs as a normal part of their operation. The SIU obtains and consolidates information about virtual entities from virtual world PDUs.

Interoperability Protocol

The SIU and the CGF Testbed communicate via ethernet using a set of PDUs called the Interoperability Protocol (IOP). SIMNET and IOP PDUs are distinguished by the arbitrarily chosen value in the type field in the association protocol layer of the SIMNET protocol. The IOP is being implemented within the DIS protocol's Data Set PDU. Earlier versions of the IOP are described in (Karr, 1994a) and (Karr, 1994b).

The currently active IOP PDUs are:

- Status Request
- Unit Detail
- Change Unit Status
- Disaggregation Response
- Operation Order
- Frag. Order
- Operator Intent
- Call for Fire Request
- Eagle Indirect Fire
- Indirect Fire Volley.

Earlier versions of the IOP had Disaggregation Request and Reaggregation Request PDUs. The process of disaggregation and reaggregation has changed since (Karr, 1994b) and these PDUs have been dropped and the Change Unit Status PDU added. The remainder of the IOP remains largely unchanged. We discuss here only the changes to the IOP and the disaggregation process since (Karr, 1994b).

Unit Detail PDU The Eagle model sends descriptions of aggregate units to the SIU which are forwarded to the virtual world in Unit Detail PDUs (UDPDU). Within each UDPDU is a field for the unit status (disaggregated, aggregate (pseudo-disaggregated), aggregate (icon), or invisible). Each UDPDU is directed to the Unit Appearance Manager (see below) which manages how units appear in the virtual environment. For example, when an aggregate unit's status changes to "disaggregated", the disaggregation process is initiated.

Two mechanisms have been developed to allow aggregate units to appear in the virtual environment. The Unit Appearance Manager stores the information in each UDPDU and produces SIMNET Vehicle Appearance PDUs (VAPDUs) at regular intervals for the aggregate units. Two "types" of VAPDUs are produced depending on the level of detail needed. The lowest level of detail, "icon unit appearance", produces one VAPDU for each unit. This VAPDU describes the unit as a SIMNET echelon and allows nodes on the net to display an icon for the aggregate unit. This approach minimizes network traffic but allows nodes in the virtual world to know about the aggregate units being simulated by Eagle. (Entity State and Aggregate Descriptor PDUs are used in the DIS protocol.)

A more detailed level of unit appearance is available to display the individual vehicles of aggregate units in the virtual world. In this approach, called "pseudo-disaggregation" (Root, 1993), VAPDUs for each of the vehicles within the unit are produced at regular intervals (every 5-10 seconds). The locations of these "pseudo-vehicles are based on formations which are determined by the operational activity of the unit. Nodes on the network see a formation of vehicles moving across the terrain. Because these pseudo-vehicles are not being simulated as entities within the virtual world, they can not fire their weapons, sight other entities, or receive fire. Pseudo-disaggregation provides a mechanism for putting "many" entities in the virtual world to create a realistic picture for sensor systems.

Change Unit Status Whenever the status of a unit changes a Change Unit Status message is sent to the SIU. This message acts as a "warning" to the SIU that Disaggregation Responses will soon be arriving. This is a new message and was unnecessary when only the Eagle model initiated all disaggregations/reaggregations. CGF operator initiated disaggregation/reaggregation were added for the AirNet installation (see below). Now the operator at a CGF OI can initiate disaggregation/reaggregation by "clicking" on units/vehicles visible on the plan view display. This message informs the SIU (which informs the Eagle model) that a unit has changed its status.

Testbed Communication Structure

Several constraints guided the design of the Testbed interface to the SIU:

1. Several OIs and Simulators might be needed to control the entities in a unit.
2. The disaggregation process should dynamically allocate OIs and Simulators in based on unit size.
3. Because the Testbed is undergoing constant modification, the interface must remain stable when viewed from the SIU.
4. The command and communication structure within the Testbed should follow the military structure.

There are three types of entities within the Testbed: actors, static managers, and dynamic managers. Actors are simulated vehicles or infantry. Static managers are the always present internal objects within the Testbed that implement the simulation functionality. Dynamic managers are simulation management objects that are created as they are needed.

To satisfy the design constraints, a dynamic manager called the Eagle CGF Manager was created as the single point of contact between the SIU and all active Testbed nodes. This is accomplished by allowing only the Eagle CGF Manager to send and respond to a Status Request PDU. The Eagle CGF Manager receives IOP PDUs from the SIU and passes each to the correct Testbed node and entity within that node. Similarly, all messages from Testbed entities to the SIU are directed through the Eagle CGF Manager.

When the Eagle CGF Manager receives a disaggregation request from the Unit Appearance Manager, it creates a "Command and Control" (C2) entity for that unit and forwards the disaggregation request to it. The C2 entities are dynamic managers and correspond to unit HQs/commanders. When a C2 entity receives a disaggregation request, it instantiates the vehicles at its level (e.g. command vehicles), creates C2 entities for its subunits, and sends subunit disaggregation requests to each of its subunit C2 entities.

The process of disaggregation descends through the command structure to the lowest subunit level which instantiates its vehicles and terminates the disaggregation process. This process is supported by "composition" and "formation" files which detail the composition of each unit in terms of subunits and vehicles and the locations of the subunits and vehicles relative to the higher unit's location.

Supporting the Eagle CGF Manager are four other types of dynamic managers: several CGF Interfaces, an CGF Interface Manager, a Unit Appearance Manager, and a Manned Simulator Manager. CGF Interfaces are dynamic managers through which the C2 entities communicate to the OIs controlling the individual vehicles. There is one CGF Interface for each available OI. The CGF Interface Manager is responsible for keeping track of the load on individual OIs and for allocating CGF Interfaces to C2 entities. Unit Detail PDUs are routed to the Unit Appearance Manager which produces the PDUs to display aggregate units (and their vehicles) in the virtual world (Root, 1994). Finally, the Manned Simulator Manager keeps track of the available manned simulators, allocates them to C2 entities during disaggregation, and activates each manned simulator when it is allocated. See figure 3 for a diagrammatic representation of the communication pathways among the dynamic managers.

AirNet Installation

In February 1994, this project shifted part of its efforts from addressing the technical and theoretical issues of the constructive-to-virtual interface issues to preparing for fielded use in training and analysis. The first stage of this preparation was the installation of the system at the Ft. Rucker AirNet facility.

Both the Eagle model and the CGF Testbed were modified to include more realistic models for air units and air vehicles (specifically Rotary Wing Aircraft (RWA)). In addition, the capability to include RWA simulators as elements of disaggregated units was added.

These new air capabilities were demonstrated in July 1994. In that demonstration, a trained U.S.Army helicopter crew

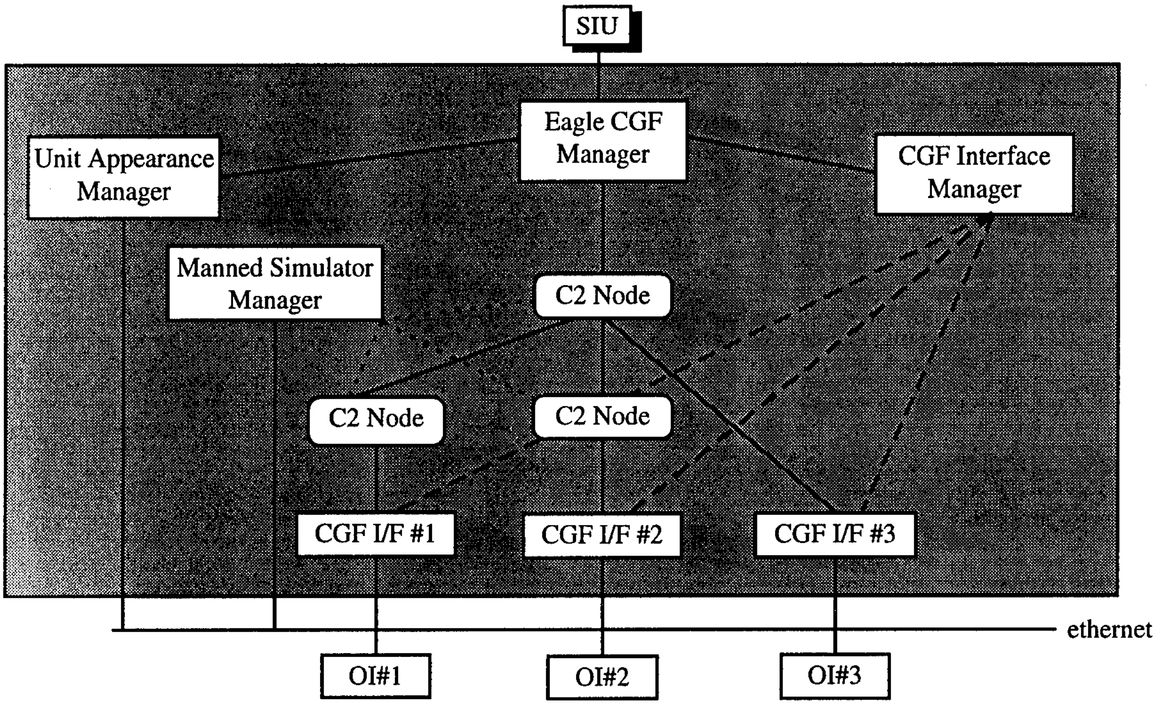


Figure 3. Testbed Communication Structure

flew a manned RWA simulator as an element of a disaggregated Eagle unit in the BDS-D environment. During their flight they encountered an Eagle controlled OPFOR unit. As the RWA approached this unit, the unit's state was changed to "pseudo-disaggregated" which allowed the crew to "sight" individual vehicles within the OPFOR unit.

As the manned RWA approached its objective, another Eagle unit, the unit of disaggregated. The RWA crew determined an engagement location for their aircraft and for two CGF controlled Apaches and radioed the location to the CGF operator. After the CGF Apaches arrived In position, the manned RWA initiated an attack on the OPFOR unit. Together, the manned RWA and the CGF Apaches destroyed the OPFOR unit.

This demonstration showed how the Eagle/BDS-D system could be used in training scenarios. Work is continuing on improving and expanding the capabilities associated with RWA simulation in preparation of its use in analytic studies.

Preliminary Results

Although a final demonstration of this project has not yet occurred, preliminary demonstrations of all components of this project have been successful.

During a typical scenario, Eagle aggregate units have moved into disaggregation areas and been disaggregated into a mixture of manned simulators and virtual vehicles under control of Testbed systems. Combat occurs in the constructive and virtual worlds. During combat, constructive world indirect fire in response to Call for Fire Requests appears in the virtual world and damages virtual vehicles. Virtual world indirect fire is communicated to Eagle where damage is assessed. Operation Orders, Frag Orders, and Operator Intent messages are sent between Eagle and the human CGF operator. Constructive units appear in the virtual world as unit icons or "pseudo-vehicles". Finally, units are reaggregated in response to reaggregation requests.

Conclusions

The preliminary results of this project are encouraging and provide solutions in four areas. First, a scheme for constructive/virtual interoperability has been developed. Second, a network protocol has been implemented which transfers command and control information as well as information detailing the activities of the entities on either side of the constructive/ virtual boundary. Third, issues of temporal and spatial correlation have been addressed and limited solutions provided. Fourth, mechanisms for interaction (specifically combat) across the constructive/virtual boundary have been implemented.

This project demonstrates the feasibility of integrating the operation of constructive and virtual simulations.

Acknowledgment

This research was sponsored by the US Army Simulation, Training, and Instrumentation Command as part of the Integrated Eagle/BDS-D project, contract number N61339-92-K-0002 and by the US Army TRADOC Analysis Center. That support is gratefully acknowledged.

References

Franceschini, R. (1992). Intelligent Placement of Disaggregated Entities. Proceedings of the 1992 Southeastern Simulation Conference (pp 20-27). Pensacola FL. Oct. 22-23 1992.

IST-CR-93-15 (1993). Proposed IEEE Standard Draft Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications Verions 2.0 Third Draft, May 28, 1993.

Karr, C., Franceschini, R., Perumalla, K., and Petty, M. (1992a). Integrating Battlefield Simulations of Different Granularity". Proceedings of the 1992 Southeastern Simulation Conference (pp 48-55). Pensacola FL. Oct. 22-23 1992.

Karr, C., Petty, M., Van Brackle, D., Cross, D., Franceschini, R., Hull, R., Provost, M., and Smith, S. (1992b). The IST Semi-Automated Forces Dismounted Infantry System: Capabilities, Implementation, and Operation (Technical Report IST-TR-92-6). Institute for Simulation and Training, University of Central Florida, 183 pages.

Karr, C., Franceschini, R., Perumalla, K., and Petty, M (1993). Integrating Aggregate and Vehicle Level Simulations. Proceeding of the Third Conference on Computer Generated Forces and Behavioral Representation (pp231-239). Orlando, FL, March 17-19, 1993..

Karr, C. (1994a). Integrated Eagle/BDS-D Interface Report (Technical Report IST-TR-94-06). Institute for Simulation and Training, University of Central Florida, 38 pages.

Karr, C. R. and Root, E. D. (1994b). Integrating Aggregate and Vehicle Level Simulations. Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation (pp425-436). Orlando FL, May 4-6 1994.

Mastaglio, T. (1991)Networked Simulators and Computer-Supported Wargame Simulations. Proceedings of the 1991 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 1 (pp. 303-307) Charlottesville VA, Oct. 13-16 1991.

Pope, A. R. (1991). The SIMNET Network and Protocols (Report No. 7102,). BBN Systems and Technologies, July 1989, 160 pages.

Powell, D. R. & Hutchinson, J. L. (1993). Eagle II: A Prototype for Multi-resolution Combat Modeling. Proceedings of the Third Confernence on Computer Generated Forces and Behavioral Representation (pp 221-230). Orlando FL, March 17-19.

Root, E. D. and Karr, C. R. (1994). Displaying Aggregate Units in a Virtual Environment. Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation (pp 497-502). Orlando FL, May 4-6 1994.

Smith, S., Karr, C., Petty, M., Franceschini., R., and Watkins, J. (1992). The IST Semi-Automated Forces Testbed (Technical Report IST-TR-92-7). Institute for Simulation and Training, University of Central Florida.

Sudkamp, T. A. (1988). Languages and Machines, Addison-Wesley Publishing Co., New York NY.

Thorpe, J. A. (1987). The New Technology of Large Scale Simulator Networking: Implications for Mastering the Art of Warfighting. Proceedings of the 9th Interservice/Industry Training Systems Conference (pp. 492-501) Orlando FL, November 30-December 2 1987.

TRAC (1993). EAGLE/BDS-D Intelligent Preprocessor User Documentation, Vol. 1.
TRAC (1993). EAGLE/BDS-D Software Requirements Document and Architectural Overview, Vol. 2.
TRAC (1993) "EAGLE/BDS-D Initial Actors/Objects in Eagle Model", Vol. 3.
TRAC (1993) "EAGLE/BDS-D Knowledge Bases and Object Classes", Vol. 4.
TRAC (1993) "EAGLE/BDS-D Detailed Design Document for Div/Corps Level Model", Vol. 5.
TRAC (1993) "EAGLE/BDS-D Eagle Concept Papers", Vol.6.
TRAC (1993) "EAGLE/BDS-D Eagle Model Inline Documentation", Vol. 7.

CONSTRUCTIVE TO VIRTUAL SIMULATION INTERCONNECTION FOR THE SOFNET-JCM INTERFACE PROJECT

CDR Dave Babcock, Maj. Jim Molnar, Maj. George Selix, Mr. Glen Conrad, Mr. Mark Castle, Mr. Jim Dunbar, Mr. Steve Gendreau, Mr. Tony Irvin, Mr. Mike Uzelac, Ms. JoAnn Matone
Martin Marietta Information Systems
Orlando, Florida, U.S.A.
June 24, 1994

ABSTRACT

Under the sponsorship and direction of the Joint WarFighting Center (JWFC) at Hurlburt Field, Special Operations Command (SOCOM) at McDill AFB, the 58th Special Operations Wing (SOW) at Kirtland AFB, and the Department of the Air Force Headquarters, Ogden Air Logistics Center (AFMC) at Hill AFB, Martin Marietta has developed a Distributed Interactive Simulation (DIS) compliant node which links the existing SOFNET virtual simulator network with a theater level constructive simulation, Joint Conflict Model (JCM). The JCM simulation and the DIS compliant interface for the JCM simulation has been developed by Lawrence Livermore National Laboratory (LLNL)/Department of Energy(DOE).

On 8 June 1994, the SOFNET-JCM Interface Project performed a highly successful, long haul demonstration between Kirtland AFB and Hurlburt Field. This Proof of Principle demonstration culminated a fast paced, nine month development effort. The objective of the demonstration was to interconnect a virtual simulator network with a joint, theater level constructive simulation. The virtual simulation network (SOFNET), located at the 58 SOW consists of three high fidelity helicopter simulators (MH-60G, MH-53J, and TH-53A) and the Training Observation Center (TOC). The project introduced a new, developmental node, the Network Interface Unit (NIU) which connects the existing network to the outside world. The constructive simulation was the JCM simulation which is used by the JWFC to conduct JTF and theater level training exercises. The communications between the simulator network and the constructive simulation were implemented via a T1 line and used the DIS protocol (V2.03) to control the data interfaces. This paper describes the architecture and design which supported this demonstration.

AUTHORS

CDR Dave Babcock, Joint WarFighting Center: CDR Dave Babcock is the JWFC's Program Manager for the SOFNET-JCM Interface Project. He is responsible for the overall management of the project.

Maj. Jim Molnar, Joint WarFighting Center: Maj. Jim Molnar is the JWFC's Program Manager for the JCM simulation. He is responsible for managing the integration of this package for the SOFNET-JCM Interface Project.

Maj. George Selix, 58th Special Operation Wing: Maj. George Selix is the Chief, Rotary Wing Training Systems Division, 58th SOW, Kirtland AFB, NM.

Mr. Glen Conrad, MITRE: Glen Conrad is a Systems Engineer with the MITRE Corporation, where he provides technical support for distributed simulation projects at the JWFC.

Mr. Mark Castle, Martin Marietta: Mark Castle is the Program Manager for external networks for the Martin Marietta Flight Systems organization.

Mr. Jim Dunbar, Martin Marietta: Jim Dunbar is the networks Chief Engineer for the Martin Marietta Flight Systems organization.

Mr. Steve Gendreau, Martin Marietta: Steve Gendreau is the lead Software Engineer for the NIU for the SOFNET-JCM project.

Mr. Tony Irvin, Martin Marietta: Tony Irvin is a Software Engineer for the SOFNET-JCM Interface Project with over five years experience in the virtual reality industry.

Mr. Mike Uzelac, Lawrence Livermore National Laboratory (LLNL): Mike Uzelac is the LLNL Project Leader for the Joint Conflict Model (JCM). He has held positions supporting development of combat simulators for training and analysis at LLNL for ten years.

Ms. JoAnn Matone, Lawrence Livermore National Laboratory (LLNL): JoAnn Matone is a Software Engineer working for the Conflict Simulation Laboratory at LLNL.

CONSTRUCTIVE TO VIRTUAL SIMULATION INTERCONNECTION FOR THE SOFNET-JCM INTERFACE PROJECT

CDR Dave Babcock, Maj. Jim Molnar, Maj. George Selix, Mr. Glen Conrad, Mr. Mark Castle, Mr. Jim Dunbar, Mr. Steve Gendreau, Mr. Tony Irvin, Mr. Mike Uzelac, Ms. JoAnn Matone
Martin Marietta Information Systems
Orlando, Florida, U.S.A.

INTRODUCTION

The Defense Science Board (DSB) 1992 Summer Study provided areas of technology that were recommended for technical demonstration in the near future to better meet the Armed Services current and future training and testing requirements. Under the Advanced Distributed Simulation (ADS) program, the process of soliciting demonstration proposals from the Unified Commanders-in-Chiefs (CINCs) Services and DoD Agencies was conducted in order to match the recommended DSB demonstration areas against operational needs. The ADS program is managed by the JWFC, Hurlburt Field, FL under the direction of the Joint Staff. As was brought out in the DSB study and numerous demonstration proposals, the capability to conduct joint training across a seamless, synthetic battlefield environment which integrates virtual, constructive, and live simulations is a vision shared by the joint modeling and simulation community and warfighters alike. One Joint Staff approved ADS demonstration proposal that emerged from the formal joint review and subsequent ADS coordinating efforts was the integration of a high fidelity virtual SOF Inter-Simulator Network (SOFNET) aircraft simulator, and a theater level constructive simulation, the JCM.

Operational goals of this ADS project are to demonstrate and evaluate the achievable interactions between the existing SOFNET virtual simulators and JCM for service and joint training applications. Interoperability between an existing high resolution "operator-in-the-loop" SOF aircraft simulator system and a theater level war game will provide unique training and readiness opportunities in the area of aircrew training, mission planning, and command and control coordination. Furthermore, it will enable CINC/Joint Task Force (JTF) staffs and SOF aircrews to perform mission review and rehearsal dynamics and coordination training within the context of a war game or "real world" event. The specific project operational goals are:

- To enhance CINC/JTF staff and supporting staff coordination training (SOF command & control, contingency planning, C4I, etc.) while conducting critical, pace setting, simulator flown SOF missions under "real-time" constraints against an interactive, simulation driven enemy during joint computer assisted exercises and training events;
- to increase scope and fidelity for training and mission review/rehearsal dynamics of CINC/JTF staffs and SOF aircrews during simulator driven "real world" rehearsal events;
- to improve SOF play in joint theater level simulations with insertion of "operator-in-the-loop" and weapons system (platform level) responses, and;
- to provide a testbed for virtual to constructive simulation interface development in a joint operational training environment.

Technical goals included demonstrating and evaluating a shared synthetic battlefield across a distributed communications network using DIS technology with existing virtual and constructive training tools. The projects focus on close interactions between the virtual and constructive simulations provides insight on issues involving what level of simulation technology is adequate for quality training with dynamic interactions between operations personnel and the command and control staff.

The principal project goal was to demonstrate that a constructive simulation could be an active player among virtual simulators and an integral component of a joint mission rehearsal environment. To accomplish this goal, the DIS network protocol was selected to connect the JCM at Hurlburt Field, Fort Walton Beach, FL to the SOFNET virtual simulators at Kirtland AFB, Albuquerque, New Mexico.

The initial project demonstration which was conducted on 8 June 1994, highlighted the potential joint training applications provided by transitioning emerging technology of interoperability between virtual and constructive simulations to an operational end-user training environment.

This ADS demonstration project is a cooperative effort between the JWFC/Joint Staff (J-7), the USAF 58 SOW/Air Education and Training Command (AETC), the LLNL/DOE, and Martin Marietta Information Systems.

SOFNET Network Architecture

Existing Network and Simulators

SOFNET is comprised of three heterogeneous, virtual, "operator-in-the-loop", helicopter simulators and a TOC. The network is designed to allow training instructors on a variety of system configurations ranging from Computer-Based Training (CBT) in the TOC's electronic, multi-media classroom, to hands-on training in a MH-60G, MH-53J, or TH-53A cockpit, or networked together via Shared Common RAM Network (SCRAMNet) memory for mission rehearsal.

Hardware Implementation

Each network node is comprised of a 6U VME chassis, a 512 Kbytes SCRAMNet memory board, VMEbus agent bus-link board, two VMEbus bus-link repeater boards, and two single board computers (SBC). SCRAMNet, developed by SYSTRAN Corp., is a real-time, fiber optic based, reflective memory device utilizing an insertion ring network topology to provide a deterministic mechanism for interprocessor communication. The agent bus-link board is the subordinate board in a two board set. Its counterpart, the host bus-link board, resides in the Helicopter Flight Simulator (HFS) chassis. Together the host and agent bus-link boards allow the agent to push and pull data to a non-contiguous 16 Meg section of HFS memory. The two bus-link repeater boards allow the Electronic Warfare (EW) system access to the HFS and to the SCRAMNet network. The first repeater board overlays the VME address space associated with the agent bus-link board providing the EW system with a contiguous 16 Meg window of HFS memory. The second repeater board overlays to 512 Kbytes of memory associated with SCRAMNet memory providing the EW system with network access. The SBCs provide the computational platform required to transfer data between the HFS and EW systems and other trainer's HFS and EW systems.

Network Software Implementation

The software resident on the two SBC, provide asynchronous access of over 100 Kbytes of real-time data to eight network players. Functionally, the software can be decomposed into five tasks.

1. Internal State processing
2. Moving Model Selection and Transfer
3. External State processing
4. Master/Slave Data Transfers
5. EW Data Transfers

More simply, these five tasks can be grouped into two primary functions. The first function manages the HFS-SOFNET interface, and is composed of tasks 1 and 2. Internal State processing ensures conductivity, availability, and timely processing between the HFS and SOFNET across the bus-link. Moving Model Selection and Transfer donates the HFS ownship and its six slewable models to the network and employs a selection algorithm designed to select the six closest slewable models, based on slant range from the local trainer's ownship, from the network and transfer the selected model data to HFS. The second function manages the SOFNET-SOFNET node interface, and is comprised of tasks 3, 4, and 5. External State processing ensures conductivity, network player participation, and network status. Master/Slave Data Transfers gives the network master the capability to distribute control of system level functions among the network players. When a network player has acquired control of a system function, the local host simulator is responsible for updating the network with current information; otherwise, the local host reads the current information from the network. EW Data Transfers opens a communication path to/from the Instructor/Operator Station (IOS) and HFS to the Master/Local EW system for command and control data.

Joint Conflict Model (JCM) Simulation

Existing Capabilities

The JCM system is a suite of software that includes tools for scenario development, the JCM simulation, and post-processing tools.

The JCM simulation is a stand-alone, event sequenced, object oriented, stochastic simulation created to support the exploration of the relationships of combat and tactical processes. JCM can operate in both interactive and batch modes to function as a training simulation system and as an analytic tool. JCM contains models of combat systems, the battlefield environment, and systems interact with each other and with the environment. In the interactive mode, the simulation processing is at or near real-time while servicing up to 24 workstations with up to two JCM players per workstation. Interactive, color graphics based, command and control planning functions allow the JCM player to create operational battle plans on the workstation while the simulation is running and to transmit these to any other workstation in the exercise in the form of graphic overlays. In the batch mode JCM runs in an ungoverned state to complete as many replications in as short a time as possible, limited only by the capability of the host computer.

The JCM scenario development uses a suite of color graphics aided, user friendly utility tools that allow the scenario developer to directly control all simulation data. These tools promote the rapid set-up, review, and change of simulation data with minimal staffing requirements.

The JCM post-processing tools provide a color graphics user interface to permit near real-time interactive analysis of simulation and exercise collected data. These tools enable the analyst to perform rapid analysis of current and past exercises thereby gaining insights in minutes that in the past may have taken months.

Software Architecture

The JCM system employs an object oriented data driven design. Over time, this design has proven to be robust and extensible. Through thoughtful data manipulation the scenario developer can simulate new and emerging systems that are not explicitly modeled in JCM without requiring code modifications.

The JCM system employs a distributed graphics architecture so that all graphics processing is performed on the workstations thereby greatly reducing the computational load on the host computer.

Computational Platforms

JCM runs on a DEC VAX host with a minimum of 32 mega bytes of memory, up to 24 Tektronix 4225 graphics terminals, and one VT 100/200 compatible terminal. Host performance requirements depends on application.

SOFNET JCM Requirements

Basic Infrastructure Requirements

Project Phase I interoperability is accomplished using five DIS Version 2.03 Protocol Data Units (PDU's) or message formats. Specific PDU's defining the Phase I interface messages are the Entity State PDU (ESPDU), the Fire PDU, the Detonate PDU, the Action Request PDU, and the Action Response PDU. The ESPDU and the Detonation PDU are used as described in the DIS specification. The Fire PDU is used as described in the DIS specification with some interpretation on the data required. The Action Request PDU and Action Response PDU provide a prototype transfer of entity modeling control. The extent of the data contained by these PDU's is limited to the Phase I requirements and may not include data for all PDU fields.

Entity Visualization and Interaction

Entities modeled in the SOFNET system are represented in JCM and vice versa. JCM entities are rendered in the SOFNET visual systems. SOFNET entities are displayed on the JCM operator(s) workstation as normal JCM entities. Entity interactions take place between and within the local simulation. Entity State data is presented on the network at the resolution of the simulation controlling the entity.

Weapons Fire Visualization

Weapons fire between JCM entities can be seen in the SOFNET simulator's visuals. Weapons fire from JCM entities directed at SOFNET simulators can be seen in the simulators' visuals. Weapons fire visual effects are instantiated automatically based on Fire and Detonation PDU traffic. For Phase I, SOFNET maps weapons fire data into three types of visual effects based on weapon type. These three visual effects are tracer burst fire, gun muzzle smoke, and in-flight missiles. Missiles fired in JCM are instantiated as JCM entities and generate Entity State PDU's during flight.

Weapons Effects

JCM weapons effects against JCM entities either destroy the target or leave the target undamaged. Destroyed JCM entities remain in the scenario, continue to generate Entity State PDU's, and are seen in the SOFNET visuals. JCM entities may "damage" the SOFNET manned simulators. JCM weapons effects against the SOFNET simulators are modeled through an "operator-in-the-loop" procedure triggered by Detonate PDU's. A common visual image of destroyed entities is provided in the SOFNET visual system.

Entity Dead Reckoning and Kinematics Smoothing

The SOFNET system dead reckons and smoothes entity kinematics received from JCM to provide a more realistic appearance of higher order kinematics modeling. JCM does not dead reckon or smooth entity kinematics data from SOFNET since that data is provided at update rates greater than the internal JCM update rate.

Prototype Entity Control Transfer

JCM and SOFNET provide a prototype capability to transfer modeling control between JCM's and between a JCM and SOFNET.

A separate prototype capability to "mount" one entity on another is associated with the control transfer. The "mount" capability allows the receiving simulation to associate the transferred entity's state data with another locally-modeled entity. In the Phase I demonstration, the "mount" capability allows SOFNET entities to transport transferred JCM entities in the Phase I scenario.

Entities transferred between JCM's are instantiated in the receiving JCM as normal JCM entities. SOFNET cannot presently instantiate independent models for transferred entities.

In the Phase I demonstration, control transfer and the "mount" capability are used with infantry-squad entities that are transported via (SOFNET) helicopter. The "transporting" helicopter's position data is used as the transferred entities position data thus providing a simple method by which to "model" the infantry team's location when under SOFNET control.

Network Topology and Communications Capabilities

Figure 1 shows the Phase I communications network. This network comprises the SOFNET ring network, a SOFNET ethernet segment, portions of the JWFC local exercise network, and an encrypted, T1 bandwidth (1.544 Mbits/sec) circuit between Hurlburt Field and Kirtland AFB. This circuit is channelized to provide 384 Kbits/sec of bandwidth for video teleconferencing and the remaining bandwidth (after circuit and bridge overhead) for simulation data.

The project network includes two JCM host computers. One JCM host is located at the SOFNET facility and one is located at the JWFC. Both JCM host computers can provide constructive simulation entities to the SOFNET. Both JCM hosts can be operating concurrently and controlling different simulated entities thereby distributing the JCM work load. The JCM host in the SOFNET facility is a smaller capacity computer and cannot provide as large a force structure as the host at the JWFC. The JCM host at the SOFNET facility provides an on-site testing machine and a backup machine in the event of leased communications failure.

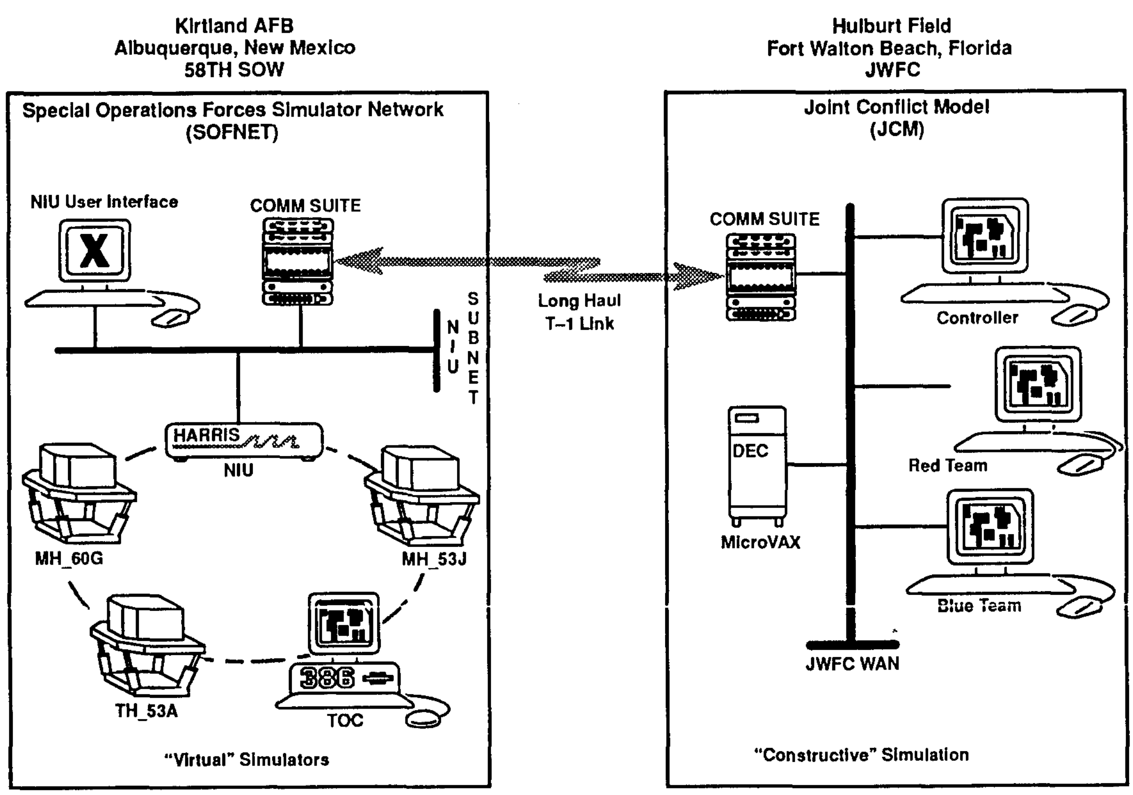


Figure 1. System Architecture

SOFNET-JCM DESIGN

SOFNET Network Interface Unit (NIU) Design Approach

The design goal for the NIU was to build a "generic" DIS interface unit which could be adapted to a variety of single simulators, Local Area Networks (LAN's), or real-time networks. To accomplish this goal, the design approach for the NIU had to be modular, portable, and easily modifiable and maintainable. For these reasons, the design team selected an object oriented design approach coupled with an Ada development environment to provide a portable, strongly typed, interface definition between modular components.

NIU Software Architecture

Decomposition of the NIU software architecture begins with identification of all the external hardware interfaces, as depicted in Figure 2. Together these interfaces give the NIU the capability of receiving DIS packets from the Wide Area Network (WAN) via ethernet, executing at real-time clock speeds, saving PDU's for off-line data analysis, participating as an active player on the SOFNET network, and interacting with the NIU operator via the X-terminal. Each hardware device has been encapsulated as a low-level object within the software design to ensure minimal impact when porting between hardware platforms.

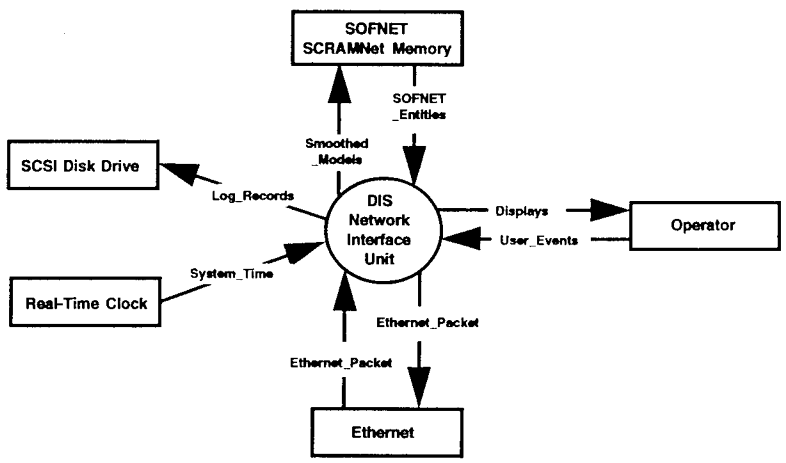


Figure 2. DIS Network Interface Unit

The next level of decomposition is derived from the natural progression data through the NIU. As depicted in Figure 3, data is symmetrically centered around the dead reckoning process and its associated entity database. Modeling this symmetry in software, an Ada "generic" Dead Reckoner was instantiated for incoming PDU traffic and another for outgoing PDU traffic. The decision to segregate Dead Reckoners solves a number of design concerns, such as task synchronization bottlenecks, entity database search duration's, flexible data transmission rates, and data integrity issues between tasks.

Generic NIU - Data Flow

To provide a more detailed understanding of the NIU design, the data flow to and from the DIS network, as depicted in Figure 3, must be described. The incoming data flow starts with the receipt of a User Datagram Protocol (UDP) message from a broadcast transmission via the ethernet device driver. The Ethernet_Packet is then passed through a series of filtering mechanisms designed to discard unknown data packets, erroneous or unsupported PDU's, and unsupported DIS entity types. If the packet was not discarded, the PDU is converted into the NIU's core data unit, a PDU_Entity. The PDU_Entity is logged and queued by the DIS Interface to the Incoming Dead Reckoner. Activated by a real-time clock interrupt, the Dead Reckoner dequeues the new PDU_Entity and modifies the entity database by either creating a new entity, updating the "truth" position of existing entity, or deleting an entity. After the Incoming_PDU_Queue is emptied, the Dead Reckoner cycles through the entity database dead reckoning all the entities to current time and passes updated PDU_Entity to the SOFNET Interface.

Looking at the outgoing data flow from a Dead Reckoner viewpoint, the processing and data flows are exactly as described above; however, the input and output subprograms used by the Dead Reckoner are different. An outgoing entity originates on SOFNET and is synchronously formatted into a PDU_Entity and passed to the Dead Reckoner. The Dead Reckoner modifies the entity database, applies the dead reckoning algorithm, and sends the updated PDU_Entity to Interface SOFNET for broadcast ethernet transmission to the DIS network.

SOFNET Interface

Creating a multi-threaded, real-time, asynchronous interface to an existing, unchangeable, real-time network, posed a number of significant design challenges; first, preserve the real-time thread for

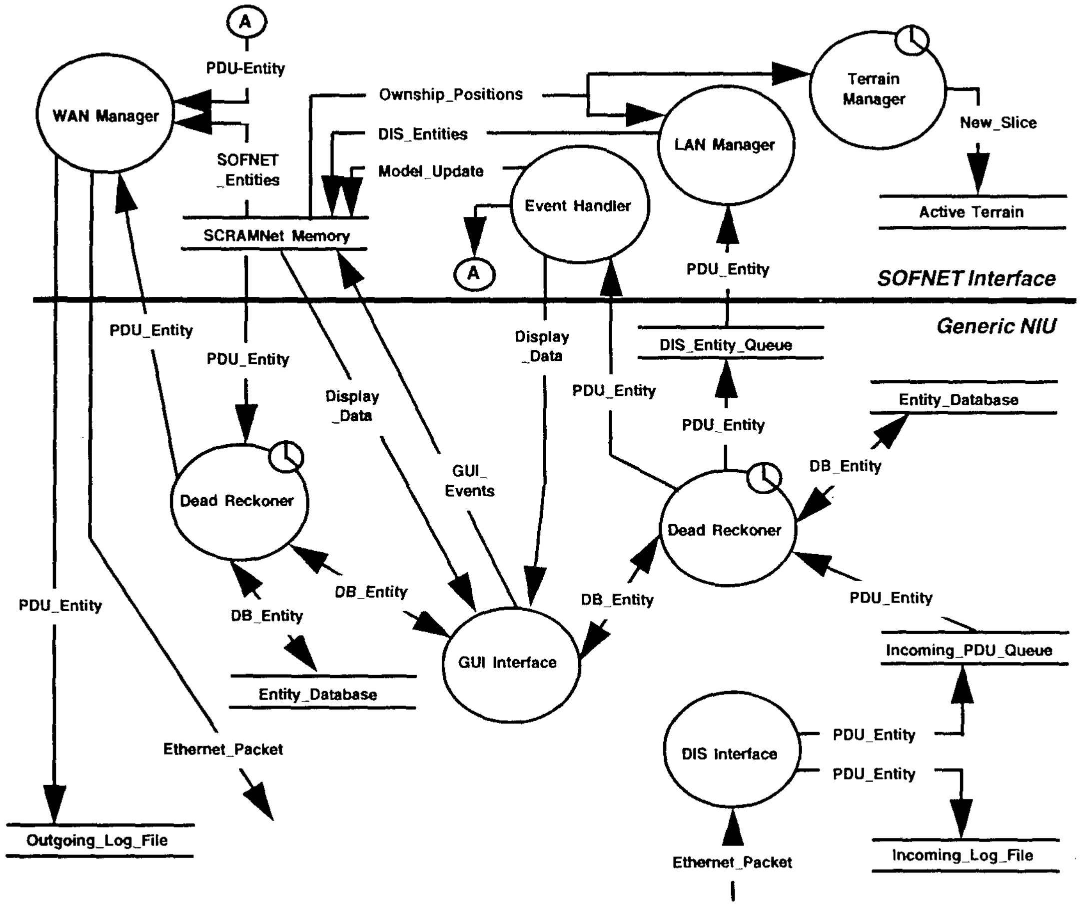


Figure 3. NIU Data Flow

incoming entities. This was accomplished by creating the DIS_Entity_Queue between the Dead Reckoner and the LAN Manager, thereby, eliminating a synchronous task rendezvous for every PDU_Entity. Executing asynchronously, these tasks updated the entity's position to current time, and then range-filtered the entities to determine the seven unique and closest entities to each SOFNET ownship. The LAN Manager is then synchronously notified by the Dead Reckoner when all entities in the database have been dead reckoned. Upon receipt of the cycle complete signal, the LAN Manager cycles through the range-filtered entities, converting their coordinates from geocentric to geodetic, translating the DIS entity type into a SOFNET moving model, smoothing the 2-D positional data into a 3-D virtual scene, correcting the entity's orientation on the database terrain, and finally updating SOFNET via SCRAMNet.

Next, the NIU ensures that the terrain surrounding each ownship is available to support terrain tracking of the DIS entities. Due to the size of the terrain data, a dynamic refresh scheme is employed reducing global memory utilization, system initialization and real-time data acquisition times. The Terrain Manager, implemented as an independent Ada task, executes cyclically, ensuring the geographic position of each ownship is centered in a 6 nautical mile box of terrain data and, if necessary, loads a New_Slice of terrain data from the disk resident terrain files.

Finally, the remaining two tasks in SOFNET Interface are event driven tasks. The WAN Manager synchronizes access to the outgoing ethernet socket, thereby shielding the socket from the multi-tasking Ada environment by single-threading access. The Event Handler is responsible for processing all special events; for example, accepting entity control, transferring entity control, and damage assessments from munition detonations.

Special Applications

As mentioned above, the special applications give the customer the ability to pick-up, transport, and unload a dismounted infantry team with a SOFNET entity, interact with a JCM ground battle, and randomly assess SOFNET entity attrition when hit by a JCM controlled munition. Integrating these special applications into the demonstration scenario significantly enhanced the realism of the SOF portion of the mission.

Dismounted Infantry Exfil

The helicopter pick-up utilized transfer of control and mount/dismount capabilities. Initiated by JCM via Action Request PDU, the Dead Reckoner removes the dismounted infantry entity, visually represented as a six-man SOF Team from the Entity_Database and sends the PDU_Entity to the Event Handler. Upon receipt, the Event Handler creates an NIU owned entity and sends an Action Response PDU to JCM accepting control of the entity. Once an entity is owned by the NIU, the entity can be mounted onto another SOFNET entity via the NIU's motif-based Graphical User Interface (GUI). Hence, to simulate the SOF Team pick-up, the NIU owned team is mounted onto the SOFNET helicopter, removed from the visual scene, and temporarily deleted from the DIS exercise. Transporting the team is accomplished by moving the helicopter and the team as a single entity. To unload the team, the NIU operator, via a GUI scrollable list, dismounts the SOF Team which returns the team to the visual scene using the helicopter current geographical position, reactivates the team as an active DIS entity, and divides the team and helicopter into two separate entities. Additionally, during the demonstration, control of the NIU owned SOF Team was returned to JCM by utilizing the same Action Request/Response acknowledgment protocol; however, the protocol was initiated by the NIU operator.

JCM Ground Battle

JCM ground battle was achieved using the Entity State, Fire, and Detonation PDU's. Simply, the NIU treated the Fire and Detonation PDU's as a create and delete message, respectively. Upon receipt of a Fire PDU, the Dead Reckoner created a new entity. If the new entity was trackable, i.e., a missile, JCM sent Entity State PDU's to update its position until the munition detonated. Upon detonation, JCM sent a Detonation PDU and the Dead Reckoner searched the entity database and deleted the entity which was created by the Fire PDU. It should be noted that if a JCM entity was destroyed during the battle, as indicated by the appearance field in the Entity State PDU, the LAN Manager would change the entity visual appearance to predefined "rubble" model.

Damage Assessment

Assessing SOFNET entity damage inflicted by JCM munitions demonstrated yet another integral piece of entity interaction. Initiated via a Detonation PDU, the Event Handler searches through the SOFNET entities looking to match the target ID from the Detonation PDU to a SOFNET entity ID. Upon a match, the damage assessment is calculated using a random number and a set of operator tunable damage probabilities to select the proper damage severity to be applied to the SOFNET entity. Then, the Event Handler modifies the SOFNET entity's appearance and notifies the NIU operator, via a GUI pop-up.

NIU Hardware Design

The NIU is a rack mounted Harris NightHawk 4400S with a Tektronics X-terminal console. The NightHawk consists of a five slot HVME/VME backplane, slots 1 and 2 are 9U HVME and slots 3, 4, and 5 are 6U VME slots. A 9U HVME CPU board, connected in slot 1, consists of two 20 MIP CPUs, 32 Meg of local memory, and a separate ethernet port used to communicate with the X-terminal. Slot 2 is occupied by a 64 Meg global memory board which is attached to the CPU board and communicates with the CPU across a high speed frontplane. An ethernet Eagle board, connected in slot 3, provides the ethernet interface to T1 communication hardware. Finally, the SCRAMNet motherboard is connected in slot 4 and the attached daugtherboard occupies slot 5.

TERRAIN DATABASE CORRELATION ISSUES

Both JCM and SOFNET use a terrain database to approximate the real world. SOFNET terrain databases use a mesh of 3-D polygons to represent terrain, while JCM uses an array of elevation grid posts for their terrain database (see Figure 4). Both databases are built from DMA DTED data, but because database formats differ and SOFNET terrain database are enhanced using map data, correlation between the two databases posed a problem with altitude and orientation for ground based entities. If JCM entity altitude did not correlate with SOFNET terrain, ground based entities would be visualized by SOFNET simulators either under ground or floating above ground.

Correlated Environmental Approaches

The first approach to solve terrain correlation was to back transform the SOFNET terrain database to a JCM database with increased elevation grid post density. Back transformation would improve database correlation, but due to the different database designs could not guarantee exact elevation correlation to the SOFNET terrain database. Realizing this, the NIU was required to place JCM ground based entities at the proper elevation and orientation to correlate with SOFNET terrain.

Because the number and type of units that JCM supports is much richer than the DIS enumeration's, the scenario developer must enter the best enumeration for each JCM unit type. In addition, the developer must enter parameters for drop interval, update rate, and broadcast information.

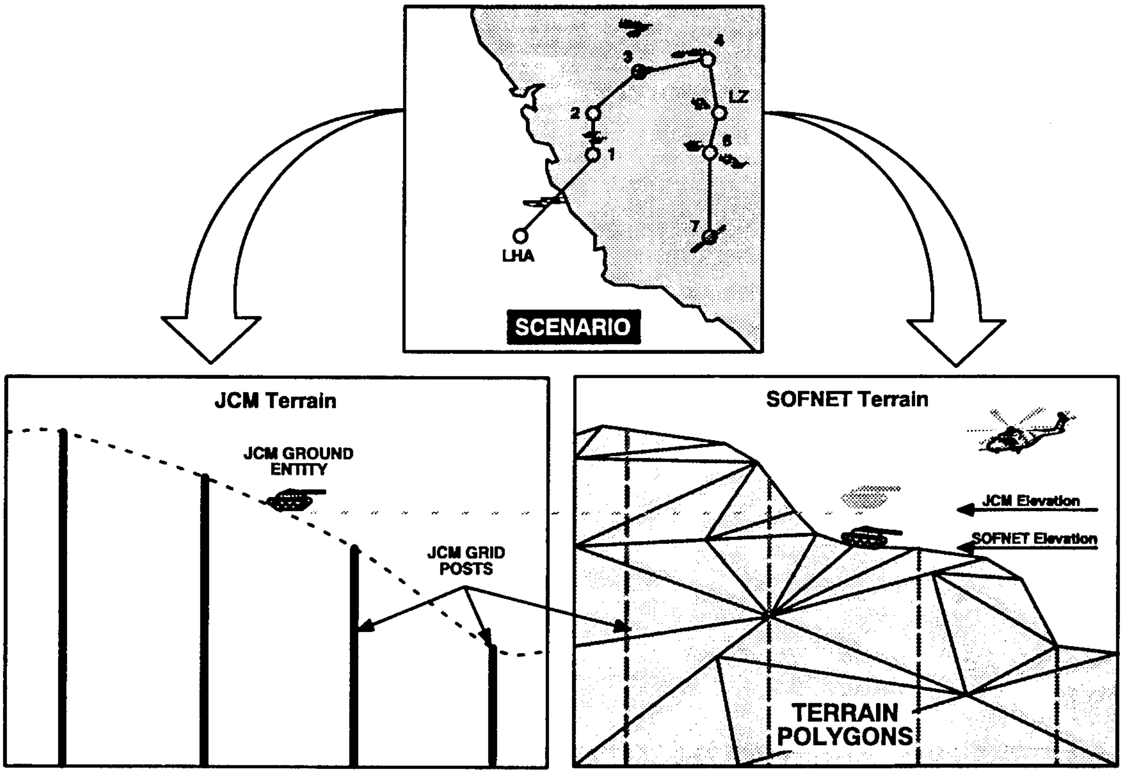


Figure 4. Terrain Correlation

Terrain Following Algorithm Incorporated into SOFNET NIU

NIU terrain following is accomplished by calculating entity altitude and orientation directly from the terrain polygon database. Using the SOFNET terrain database insures that entity elevation and orientation will correlate exactly to the visual terrain. Terrain following is accomplished by using the Terrain Manager to maintain continuous terrain data on-line while the LAN Manager calls the terrain following algorithm to calculate entity elevation in real-time.

JCM DESIGN APPROACH

JCM Modifications

Scenario Editor

The Scenario Editor (see Figure 5) allows the scenario developer to construct and review scenario data.

JCM Simulation

The JCM simulation, supports up to 48 forces distributed among up to five sides. Each force can be assigned to a graphics workstation where the force can be gamed by up to two JCM players. The JCM player controls how the JCM game entities or units move, look, and shoot. In a DIS exercise, a ghost force is assigned to each side. A ghost force is a force than can be acquired and targeted but cannot be controlled because they are under another DIS player's control. When an Entity State PDU is received for an entity gamed by another DIS player, it is added to the appropriate ghost force.

JCM DIS Interface Package (DIP)

The JCM DIP is embedded into JCM as a multi-threaded process. The DIP accepts event messages from JCM, translates the messages into DIS PDU's and broadcasts the PDU's to the DIS community. The

DIP also accepts DIS PDU's from the DIS community, translates the PDU's into JCM event messages and submits the messages to JCM for processing.

JCM and the DIP interact with each other through a series of event messages in three data stores: the Send Event Buffer, the Receive Event Buffer, and the Entity State Store. The Send Event Buffer enables JCM to post event messages directing DIP to send the on demand PDU's (i.e., Fire, Detonate, Action Request, and Action Response). The Receive Event Buffer enables DIP to post event messages which request processing by the JCM simulation. Types of processing include creating a new unit, dropping a nonresponding unit, scheduling an impact assessment, and accepting or acknowledging an entity transfer. The Entity State Store collects information about the JCM units from JCM to be broadcast via the Entity State PDU and about the DIS entities from the Entity State PDU to be ghosted by JCM.

DIP is responsible for translating DIS information into JCM format and JCM information into DIS format. Three types of translations are supported: unit conversions, coordinate conversions, and entity conversions. Entity conversions translate the DIS entity identification and entity type into a unique JCM unit of the desired unit type. JCM maintains a complete history of the exercise; therefore, it is critical for JCM to maintain a unique mapping for each DIS entity to its equivalent JCM unit. If the entity stops broadcasting PDU's and later resumes broadcasting PDU's with the same entity ID, JCM must recognize that this is not a new entity. If control of an entity is transferred to another DIS player, JCM must be able to associate the new entity ID to the old entity ID and resolve the entity to its original JCM unit.

FUTURE APPLICATIONS

Future development of the SOFNET-JCM Interface project may involve networking to other simulators, simulations, and command and control networks for enhanced training applications. Additionally, the SOFNET-JCM Interface Project Phase I efforts will provide a complimentary test bed for other DIS applications such as Modular Semi-Automated Forces (MODSAF) and other constructive interfaces with the high resolution visual systems and the SOFNET facility. This virtual to constructive simulation test bed may also be used to evaluate interactions between the virtual reality helicopter gunner stations, the integrated electronic combat suite, and other DIS-compliant applications as these capabilities are incorporated into SOFNET.

CONCLUSION

The SOFNET-JCM Interface Project provides insights into the future technical requirements necessary to meet the challenge of providing simulation interoperability in order to provide quality training at various levels within the Armed Forces. Phase I of this project serves as a springboard for these future development initiatives. Mission review and mission rehearsal training are cornerstones for warfighters' readiness training. The SOFNET-JCM Interface Project transitions emerging technology to meet this requirement.

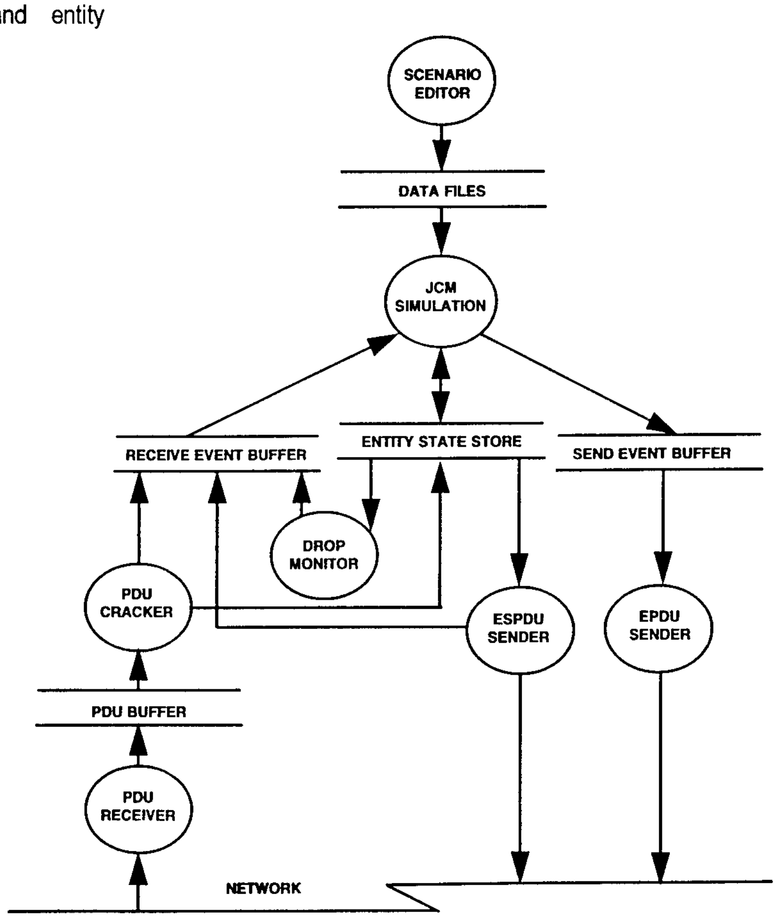


Figure 5. JCM DIS Interface Package

ACHIEVING CONSISTENT COLORS AND TEXTURES IN VISUAL SIMULATIONS

Roy Latham
Computer Graphics Systems Development Corporation
Mountain View, California 94043-2330

ABSTRACT

With increased emphasis on the verification and validation of simulations, it is increasingly important to match the colors and contrasts of the real world in the visual scene simulation. For example, in military simulations the ability to detect and recognize targets depends in part upon the colors and contrasts rendered for the target objects relative to background objects. Visual simulations are produced by digital image generators based upon polygon databases. Each polygon in the database is tagged with a color description or a texture description that ultimately leads to the appearance of the polygon in the rendered image. This paper addresses the problems of ensuring that the rendered appearance is both in accord with the real world and with other simulators. The suggested approach to achieving traceable colors involves (1) cataloging a selection of real world colors by measuring them as they occur, (2) obtaining texture patterns from photographs of real world textures either directly through image processing or indirectly by synthesizing patterns to match the characteristics of the photographs, (3) color correcting the texture images by identifying two or more colors in the pattern that also appear in the catalog of real objects and transforming the pattern colors accordingly, and (4) calibrating the image generator and display system to reproduce assigned colors as accurately as possible. Practical limitations due to the color gamuts of display systems are discussed.

ABOUT THE AUTHOR

Roy Latham is president of Computer Graphics Systems Development Corporation, a consulting firm specializing in real time graphics and simulation systems technology. Previously, he managed graphics projects at Sun Microsystems, Kaiser Electronics, and the Singer-Link Flight Simulation Division. He holds patents in simulator visual systems, has published numerous papers in the field of training and simulation, and has authored the book, *The Dictionary of Computer Graphics Technology and Applications*. Mr. Latham holds BS degrees in Electrical Engineering and in Aeronautics and Astronautics from M.I.T., an MS in Applied Mathematics from the State University of New York at Stony Brook, and an MS in Computer Science from the University of Santa Clara. He is a registered professional engineer and licensed patent agent.

ACHIEVING CONSISTENT COLORS AND TEXTURES IN VISUAL SIMULATION

Roy Latham
Computer Graphics Systems Development Corporation
Mountain View, California 94043-2330

RATIONALE FOR MATCHING

Visual images produced by different image generators for the same scenario usually look quite different. Differences in the geometry of the databases is not the main cause for the scenes looking different, although geometric differences are important. While all image generators use virtually identical calculations of geometric perspective, they differ widely in their derivations and treatments of colors and textures.

There are at least three reasons why the differences in colors and textures should be minimized:

- In military simulations, colors and textures relate to the ability to engage threats, to identify friend or foe, and to recognize landmarks. The user of a simulator with high target to background contrast will have an unfair advantage.
- If consistency among simulators were achieved without matching the real world, a tactic or system found to work well in simulation might not work well in the real world.
- Matching real world colors is the best way to achieve consistent, aesthetically pleasing colors. Otherwise database elements from different sources will not match.

Concerns with the verification and validation of networked simulation are rising as more and more questions of system design are being resolved through simulations with many players. Providing accurate colors and contrast in visual displays is one aspect of improved fidelity.

STEPS IN MATCHING

The process of matching real world colors can be broken into steps, each of which must be understood and controlled to be able to enable tracability of colors from the real objects to the computer screen. The steps are:

1. Measuring the color properties of real world objects, primarily color reflectances.
2. Representing the color properties of objects in a digital database.
3. Applying illumination and atmospheric haze effects in an image generator.
4. Calibrating the display and correcting the image to compensate for display characteristics.

There are limitations in current technology at each step that collectively prevent the overall process from achieving a perfect representation of real world colors and contrasts. This paper discusses those limitations along with the steps in the processes.

MEASURING COLOR PROPERTIES

The word *color* has the longest definitions of any word in an unabridged dictionary. The length of the definition presages inevitable semantic difficulties in discussing the subject. For example, is a tree that is green in the day time still properly referred to as being green at night, when it appears to be black? The answer is yes. In one sense of the word, *color* refers to the properties of an object that relate to selectively reflecting light. The tree retains the physical property of reflecting more green wavelengths of light than other wavelengths of light, even if the tree happens not to be illuminated.

Another sense of the word *color* refers to how an object appears, which varies with the amount and color of the illumination source, and with filters used in viewing. Both uses of *color* are valid, but when discussing the colors of objects in the real world the intent is to capture information about the physical properties of the object. If those properties are adequately measured and described in a database, then a visual simulator will be able, assuming it has the right algorithms to recreate the appearance of the object under many different conditions of illumination, atmospheric haze conditions, and possibly even simulated optical filtering.

Colors of objects in nature can be measured by matching the object color to a book of reference color chips [1], by using instruments [2], or by photography using film having known spectral sensitivity. Russian scientist E.L. Krinov collected data of many natural colors using photographic plates of varying spectral sensitivity. The data, collected from 1932 to 1942, has recently been put in modern format and newer data added. [3]

Reflectance is the primary color attribute of interest for simulation applications. A reflectance value in the range zero to one is stored for each of three color components in the simulator database. Reflectance can be adequately represented using fewer bits than that required for the intensities computed after the modeled objects are illuminated in a scene. In the real world, day scene illumination on a white cloud may yield a luminance of 10,000 ft.-lamberts. The same cloud illuminated by moonlight might yield 0.01 ft-lambert.

In the data from Krinov, the least reflective object was a fir tree measured in late summer, about 3% reflective. The most reflective was plywood, about 65%, though we know separately clouds can exceed that. A single scene can have a much higher contrast ratio than the ratio of reflectances because some objects are in shade. For example, the illumination from open sky is about 20% of the direct sun illumination. A portion of a fir tree in open shade would therefore appear less than 0.06% as bright as a cloud in sun. There is no apparent lower limit to the depth of shadow darkness, especially noting tunnels and the like. On the high end of the dynamic range, the sun or another bright illumination source may appear in an image, or, possibly specular reflections, like glint from glass.

Accommodating the dynamic range of day-to-night illumination and the dynamic range within a scene is a major challenge to image generator technology. However, the reflectances of objects do not change with illumination conditions, and remain well behaved within the range of about 3% to 95%. A texture pattern of any ordinary object can be made with eight bits per color component.

There are other properties of a surface, called appearance properties, that are important for high fidelity rendering, but which are rarely used in imagery generated in real time. The specular reflectance characteristics associated with glossy surfaces are, for example, typically omitted in visual simulation although often included in non-real-time rendering. [4]

DATABASE REPRESENTATION

Measurements of real world colors are best expressed in the units of the international CIE standards, preferably CIE xyY coordinates. Aside from the virtue of having a standardized meaning, the system also has the virtue of being capable of annotating all visible colors. Representing colors by specifying a mixture of red, blue, and green (RGB) primary colors has neither universal meaning nor annotation capability. RGB coordinates are specific to the colors of the RGB primaries for specific types of CRT (or other displays), and the phosphor colors vary. For example, the monitors used in computer workstations use different primaries from those in television sets. RGB systems, unlike the CIE xyY system, can only express colors within the gamut of the display device. For example, the pure colors of a laser cannot be reproduced by CRT phosphors, so there is no means of specifying the laser color as an RGB coordinate point. Consequently, CIE coordinates are the obvious means for representing measurement data and for interchanging the data.

CIE color representations have the disadvantage that even experienced users have difficulty intuitively relating the coordinate expressions to real world colors. The third coordinate, Y, is the reflectance scaled from 0 to 100, but x,y have no ready interpretation. White is about (1/3, 1/3), increasing x is roughly towards green, increasing y is roughly towards red, and decreasing y is roughly towards blue. The lack of intuitive interpretation makes it more difficult to check data for transcription errors and the like. It is therefore convenient to show transformed versions in Munsell space (a standard which gives hue, saturation, and value components) and a standard color name along with CIE coordinates if the colors are to be interpreted by users.

As noted, the RGB color primaries used by practical display systems vary with the type of display and sometimes among the individual devices. Run time databases must be made compatible with the primaries of the devices in use. If the color data are stored in CIE coordinates, an appropriate linear transformation (multiplying xyY by a 3 x 3 matrix) will convert the color space to the display primaries. The haze color(s) and texture colors must be transformed as well as the object colors. If the colors primaries vary among display devices, consistency will be achieved only by using different primaries for each device. The differing transformations could be computed off-line and stored for each color, or the transformation could be done as part of the polygon processing.

In principle, a 3 x 3 matrix multiplier could be included in hardware at the video output of the image generator. Doing so would allow the image generator to easily adapt to display devices having different primary colors. However, no image generator is known to use this technique.

Transforming from CIE to RGB color spaces, whether done in software or in hardware, may entail a problem of the transformed color being *out of gamut* for the RGB device. While the CIE space represents all colors, the display device cannot reproduce all colors. Highly saturated colors are inevitably the problem, which is manifested by one of the transformed RGB values being negative. An expedient cure is to take the nearest admissible color in RGB space as an approximation, where *nearest* is measured as distance in a perceptually equalized color space such as L*a*b* space. Perceptually equalized color spaces are explained in standard texts on color science. [5, 6]

Few colors in nature (vivid flowers, for example) are outside the color space of a CRT. Manmade objects such as safety markings and colored lights are more often out of the RGB gamut.

Table I provides a sample of the colors of various objects measured by the author and transformed into various color systems using commercial software [7]. The daisy and rose flowers were too saturated to be matched by an RGB monitor. Note that one of the RGB coordinates is zero, a result of taking the nearest point in the RGB color gamut.

Deriving Texture Patterns

Color texture patterns provide variations of the color reflectivities of a surface sampled at a regular grid of points over the surface. Representing each sample of the surface reflectivity poses no particular problem beyond that discussed above. Part of the pattern variations are, typically, due to shadows

Consequently, the nature of the pattern may in fact be dependent upon lighting conditions. A simple expedient is to note the lighting conditions (sun angle and direct or overcast conditions) for which the pattern was derived and to use a pattern matching those conditions in the simulator.

Even if representing the texture pattern poses no added difficulties over representing colors in general, collecting pattern data does pose challenges. The most efficient way of collecting a large array ofsampled colors for a texture pattern is to derive the pattern from a photograph. For the color samples to accurately match the real world the photograph must be corrected to match real world colors.

One starting point for correcting the colors in a photograph is to include a reference chart having known colors in the photograph. The overall brightness, contrast, and color balance of the photograph can then be adjusted so that the reference chart colors are correct and, presumably, the rest of the image is corrected at the same time. The correction process can be performed interactively using commercial image processing software or custom software can be written to support the task.

The color correction process must take into

Table I. A Sample Color Catalog

Object	Munsell Color	NBS Color Name	RGB (monitor)	xyY
Mod. weathered redwood	10YR 4.5/4	mod. yellowish brown	35, 43, 74	0.4085, 0.3891, 15.56
Flower (bract), bougainvillea	10RP 4/12	strong purplish red	97, 3, 35	0.4789, 0.2717, 12.00
Flower, daisy	5Y 8.5/12	vivid yellow	232, 174, 0	0.5035, 0.4745, 68.41
Flower, rose	5R 4/16	vivid red	110, 0, 2	0.6039, 0.2978, 12.00
Foliage, agave, leaf center	5GY 4/4	moderate olive green	27, 35, 9	0.3538, 0.4284, 12.00
Foliage, agave, leaf edge	5Y 8.5/8	light yellow	218, 176, 29	0.4117, 0.4347, 68.41
Foliage, bamboo, green	5GY 5/6	mod. yellow green	42, 60, 9	0.3663, 0.4614, 19.77
Foliage, bamboo, yellow	10Y 8/10	strong greenish yellow	168, 164, 3	0.4190, 0.4790, 36.17
Foliage, cedar tree	7.5GY 5/4	moderate yellow green	42, 52, 29	0.3274, 0.3994, 19.77
Foliage, myrtle tree	7.5GY 3.5/3	moderate olive green	19, 24, 14	0.3235, 0.3912, 9.00
Foliage, oleander	5GY 4/4	moderate olive green	27, 35, 9	0.3538, 0.4284, 12.00
Foliage, olive tree	5GY 6.5/5	moderate yellow green	91, 101, 29	0.3661, 0.4192, 36.17
Grass, lawn 1	5GY 4/4	moderate olive green	27, 35, 9	0.3538, 0.4284, 12.00
Grass, lawn 2	5GY 5/6	moderate yellow green	42, 60, 9	0.3663, 0.4614, 19.77
Paving, conglomerate	2.5Y 7/3	grayish yellow	137, 105, 69	0.3600, 0.3655, 43.06

account several practical considerations:

- A digitized photograph often contains regions where one or more of the color components became saturated in the digitization process. These regions cannot be used as reference colors.
- Digitization introduces noise, so that a single pixel in a reference color patch may not represent the whole. It is better to use averages of pixels in reference areas.
- Color film emulsions yield non-linear responses to light (particularly at the exposure extremes) and also undesired cross-coupling among the colors. Non-linear corrections are typically required.

If these factors are taken into account a suitably corrected image can be produced.

Figure 1 shows a monochomatic example of recovering reflectance values from a photographic image. The grass was photographed with a color reference chart [8] in the image and was digitized commercially as a step in Kodak PhotoCD™ processing. The photographic processing includes automatic exposure compensation that adjusts the tonal values to make a pleasant-looking image. Using the reference chart, the reflectance values of the pattern can be recovered. The grass reflects an average of only about 0.13 of the incident light, so the reflectance-recovered image is comparatively quite dark.

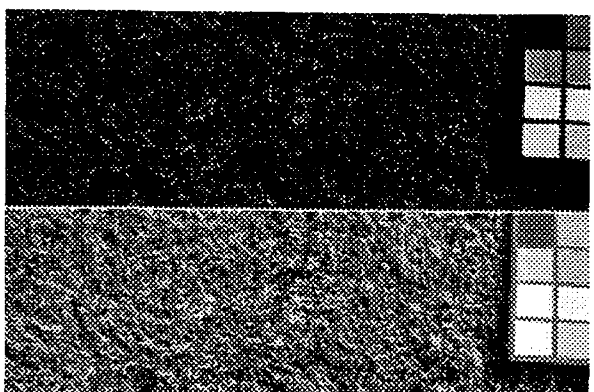


Figure 1. Low reflectance values (top) are recovered from an automatically exposed image (bottom) through inclusion of a reference chart.

Correction by Color Cataloging

In many cases, including the common one of texture patterns being derived from aerial photographs, it is not possible to include a reference color chart in the image. For such cases, a possible technique is to use *color cataloging,* a concept currently under investigation.

The concept is to identify one or more objects in the photograph that can be associated with colors measured separately. These objects of known color need not be in the portion of the photograph having the texture pattern; the whole photograph is corrected before the pattern is extracted. A typical aerial photograph might include, for example, asphalt or concrete paving, residential lawns, wooded areas of recognizable types, beach sand, snow, or sea surf. The cataloged colors of one or more of the identified objects could then be used as reference colors for the correction process.

Using natural objects for color correction will not be as accurate as using predetermined reference color patches. Nonetheless, there are reasons to suppose it may be acceptable. The colors in nature, particularly ordinary green foliage colors, fall within a fairly narrow range. Note the similarity of the foliage colors included in Table 1. Consequently, even if foliage is misidentified as an incorrect type, the error is unlikely to be dramatic. Secondly, should the process fail to exactly match the real world of the date and place of the photograph, using colors from a catalog of colors in the real world ensures that the results are at least representative of real world colors. That said, the method still bears further investigation.

IMAGE GENERATOR PROCESSING

The fundamental illumination processing in an image generator is flat shading computed according to:

$$C_i = (\underline{S} \cdot \underline{N} + D) \times R_i$$

where,

i the color components r, g, b

C_i the magnitude of a color component

$\underline{S}$ an illumination vector in the direction from an infinitely distant sun, usually with unit magnitude for day illumination

$\underline{N}$ the normal vector for the surface being illuminated

D a constant diffuse illumination, representing the illumination from the sky

R_i the reflectivity of the ith color component

The effect of this computation is to make polygonal surfaces darker when they are oriented away from the sun. However, no common image generator takes into account other objects either blocking the sun or reflecting more light onto the object. Thus a building that ought to be shadowed in a deep narrow valley nonetheless receives as much illumination as one on a hilltop. This error is usually ignored, but it can be at least partially taken into account by modeling objects in shade darker than those in sun. The shade areas are time-of-day dependent, so this leads to a need to have different databases for different times of day.

A method originated by David Hinkle at Link Flight Simulation is to darken terrain using smooth shading in valleys. The idea is to examine the terrain polygons on either side of each shared edge. If the two polygons across the edge define an acute angle, the region near the edge is shaded darker than the rest of the polygon. The narrower the angle, the darker the shading, as would occur from having less sky illumination. The effect adds realism to terrain imagery.

Having a texture capability in the image generator entails additional illumination computations. The texture pattern defines the reflectivity for each pixel of a polygon being rendered. Noting that ($\underline{S} \cdot \underline{N} + D$) is constant for the whole polygon, the polygon processing can be started with the polygon assumed to be white, all color components having a reflectivity of one, and the texture pattern reflectance values used to multiply the components for each pixel.

Multiplying the texture reflectivities works with either full color or intensity texture modulation. In the case of intensity texture modulation, a single texture value is used for each of the three color components. For intensity-only texture, the underlying polygon color may be other than white to provide a variety of monochrome patterns. The underlying face should generally be modeled at its maximum reflectance, and the texture modulation (varying between zero and one) multiplied to reduce the surface from its nominal value. Shading may reduce the intensity of the face color before texture is applied.

Non-Standard Illumination Models

The illumination computations associated with texture are sometimes performed by an image generator in ways inconsistent with standard illumination models. The pattern modulations are sometimes added rather than multiplied with the underlying pixel components, which leads to time-of-day inconsistencies with the patterns. The texture values may not be normalized to one, so that the modulation can increase the brightness of the polygons; this leads to overflow of color component values which in turn are then subject to creative means of making corrections. Sometimes texture patterns are applied so as to replace, rather than modify, the underlying polygon color.

In machines using non-standard models, the patterns may have to be changed for each illumination condition to achieve behavior that approximates real world color and textures. If the texture pattern replaces, rather than modifies, the underlying polygon color, then the illumination computation that takes into account the orientation of the pattern with respect to the sun will have to be performed independently of the image generator for each time of day and polygon orientation.

In all cases in which the image generator applies non-standard illumination models, the principle is to modify the texture pattern to compensate for the calculations made in the image generator.

DISPLAY COMPENSATION

Ideally, a visual display reproduces color brightness proportional to the image-generator-computed value for each color component of each pixel. Practical displays depart from this ideal in several ways, but one can correct some of the departures by compensation in the image generator.

Gamma Correction

The most common of these is gamma correction, which is performed near the output of the image generator to correct for non-linearities inherent in cathode ray tube displays, and some other display technologies. In principle, gamma correction could be built into the display electronics so that the display was compensated to provide brightness linearly proportional to input. However, traditionally in simulation the compensation has been performed by the image generator.

Note that broadcast television standards require a nominal gamma correction to be applied to the signal before it is transmitted, relieving each receiving set of the need for the gamma correction electronics. When broadcast video is displayed on a device that does not require gamma correction, such as an LCD flat panel, the broadcast-included gamma correction must be removed by a complementary look-up in the video circuitry prior to display. Also, if broadcast video is inset into a gamma-corrected visual system, care must be taken to avoid gamma correction being applied twice to the video inset.

Gamma correction generally cannot be performed in the image generator prior to reading the digital data for output. Gamma correction cannot be applied to the colors of the digital database because those colors are not the final ones that are output. Illumination processing changes the colors, as do texture, atmospheric haze, antialiasing, and transparency processing. Gamma correction is a non-linear mapping of each color component, so the processing within the image generator would produce incorrect results when colors are modified or combined.

For best results, gamma correction tables should be applied separately for each color component of the display. Having separate tables for each component not only allows for variations among the gamma characteristics of the display colors, it allows the gamma tables to be used to compensate for other types of calibration errors.

Display Calibration

A well-calibrated display produces no light for zero-valued input pixels, and has brightness linearly proportional to other pixel values, up to the maximum. The threshold input below which the output is zero is called the *black level,* and the ratio of output to non-zero input is called the *gain* or *contrast.* Calibration of a display includes measuring and adjusting the black level and contrast for each of the three color components.

Display calibration is best done using a photometer, an instrument that measures light levels. For black level and contrast calibration, the instrument need not measure the color components, only luminance. The calibration is performed by filling the whole display with pixels of the same input value of a single color component, with the other components kept black. Many displays vary in brightness across their surface, so all measurements should be made with the same photometer held with a fixture to keep it aimed at the same part of the image. Some instruments cannot accurately measure a small portion of a scanned display.

Stepping through each input value, typically from 0 to 255, a curve of measured output luminance is developed for each color component input value. The black level should be within about 0 to 5, the maximum output should be reached within 250 to 255, and curve smooth and continuous in between. If these characteristics are not observed, the display hardware should be adjusted or repaired. When within bounds, a gamma table can be constructed that will map input value to linear luminance from black to the maximum.

The display should be filled with a uniform intensity when making the measurements, otherwise light scattered from bright portions of the display will spoil measurements in the darker portions. Measuring the intensities of steps in a gradient scale is not accurate for these reasons, and also because display brightness is often not uniform. The color components should be done separately to allow a matching problem to be isolated to an individual hardware element associated with a display color component.

The process of stepping through input values and luminance measurements can be automated under computer control. A photometer with a digital output supplies measurements to a computer, typically the visual simulation host computer that provides input to the image generator. There is no point in fitting a curve to the data; the gamma correction table can be constructed directly from the measured data point-by-point.

Display calibration is most critical when two or more adjacent displays must be matched. Achieving good results in such cases depends critically upon using instrument methods. Adjusting display gain and contrast without instrument measurements can yield acceptable matching under one or two conditions of scene illumination, but mismatches may show up in other conditions of scene content and lighting.

Checking Display Calibration

While display calibration can rarely be accomplished satisfactorily without light measuring instruments, there are useful quick checks that can be performed without instruments. To see if adjacent displays are matching, use a continuous gradient from black to white (or gray) that is designed to match across display boundaries. By using different gradient patterns with different maximum brightnesses, the problem of the bright part of the display washing out the dark region is overcome. The intensities of the gradients should match uniformly across display boundaries. The color should remain neutral at all intensity levels.

Gamma correction can be checked using a method worked out at the Canadian Bureau of Standards in the early 1980's. Fill half the screen with a 50% gray shade, and the other half with scanlines alternately black and 100% white (Fig.2). The average brightness of the two halves will be the same if gamma correction is correct at the 50% point. Similarly, a solid 25% gray shade is checked against alternate lines of 50% intensity, and so forth for lower intensities. Alternate scanlines should be used rather than alternate pixels so that possible limitations in the frequency response of the display do not affect the outcome.

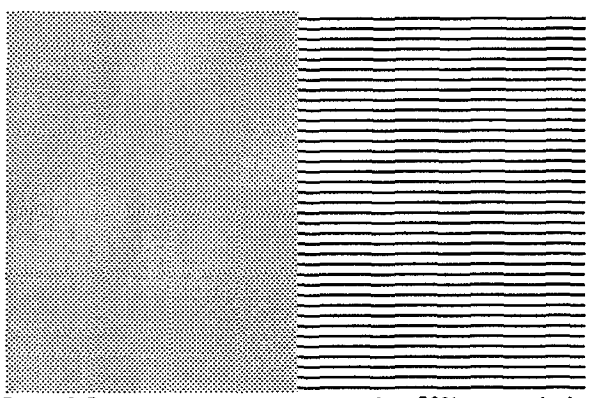


Figure 2 Proper gamma correction matches 50% gray with the average of alternating black and white lines.

SUMMARY

Achieving correct colors and intensities is a painstaking process. Critical steps are the collecting of real world color and texture data, representing the colors correctly in the databases of the image generator for the display system to be used, processing colors and illuminations with standard algorithms, and applying the corrections needed for a properly calibrated display system. All of the steps must be performed, and performed with care, to achieve accuracy and consistency.

REFERENCES

1. Munsell™ Book of Color, Munsell Color, New Windsor, New York
2. Minolta, Inc. and Kollmorgen Instruments Corporation each manufacture instruments suitable for field measurements of colors.
3. Howard, Celeste M., and J.A. Burnidge, "Colors in Natural Landscapes", Final Report AL/HR-TR-1993-0172, Air Force Human Resources Directorate, Aircrew Training Research Division, Mesa, AZ, January, 1994
4. Hall, Roy, *Illumination and Color in Computer Generated Imagery*, Springer-Verlag,, New York, 1989
5. Wyszecki, Gunter, and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, Second Edition, John Wiley & Sons, New York, 1982
6. Kuehni, Rolf G., *Color: Essence and Logic*, Van Nostrand Reinhold, New York, 1983
7. *Color Science Library*, CGSD Corporation, Mountain View, CA
8. *MacBeth ColorChecker™ chart*, Macbeth, Munsell Color, New Windsor, New York

VISIONICS DATA BASE GENERATION: AN INTEGRAL PART OF TRAINING, PLANNING AND MISSION REHEARSAL

J. Jeffrey Lombardi, Martin Marietta Flight Systems
Lt Col Edward T. Reed, 58 OG/OGU (USAF)

ABSTRACT

The 58th Special Operations Wing (SOW) of the Air Force Air Education and Training Command and Martin Marietta currently operate the largest data base generation facility in DoD tasked with producing high fidelity photo specific simulation data bases for DoD customers world wide. Started in August 1990, initial data base support was limited to five data base engineers producing basic training environments within western United States and the development of small mission rehearsal areas that were utilized by Air Force personnel only. Today, this facility has grown to twenty data base engineers and three full time intelligence personnel working around-the-clock seven days a week. Utilizing a dedicated state of the art Sun and Silicon Graphics network encompassing the latest technologically advanced applications, this team has produced nearly seven hundred thousand square nautical miles of visual data base supporting multiple customers in the Department of Defense.

This paper addresses the high fidelity simulation data base generation and the application of the standardization scheme developed at Kirtland to overcome the many challenges inherent in the construction of data bases. The joint contractor, government team at Kirtland has developed a standardization methodology that promotes efficiency, reduces cost, and improves quality. The technological barriers overcome involved integrating multiple disjointed data bases into a single contiguous landmass, converting data bases into multiple Image Generator formats, and scrutinizing the DMA specifications. The development of these standards and the substantial experience of the 58th SOW data base generation facility was instrumental in DoD's decision to co-locate the Project 2851 Simulator Data Base Facility (SDBF) at Kirtland. This facility will be networked with the 58 SOW data base facility for data base production and transformation synergy which will benefit all of DoD and industry.

About the Authors

Jeff Lombardi is the manager of the Martin Marietta Visionics Data Base Engineering Group at Kirtland Air Force Base, New Mexico. He is responsible for insuring the quality and accuracy of high fidelity visual and associated correlated support data bases. Previously, Jeff was the Technical Director of this same data base group and has been involved in visual data base generation supporting simulation and mission rehearsal activities for over 4 years. Jeff holds a B.S. in Electrical Engineering from Colorado State University.

Lt Col Edward T. Reed is Director, Aircrew Training Systems, 58th Operations Group, Kirtland Air Force Base, New Mexico. He is responsible for the management and operations of both fixed and rotary wing aircrew training devices which support the wing's special operations and air rescue training missions. He has been qualified in a variety of H-53 variants including HH-53B, HH-53C, HH-53C (NRS), MH-53H, and MH-53J, serving in both Combat Rescue and Special Operations Squadrons. His previous duties have included: Special operations program manager at HQ MAC, Scott AFB IL, HQ USAF, and the Assistant Secretary of the Air Force for Acquisition Staff Washington, DC, Chief Combat Tactics 1 SOW, Hurlburt Fld FL; MH-53H Instructor Pilot, 20 SOS, Hurlburt Fld FL; and HH-53C Aircraft Commander, 41 ARRS, McClellan AFB CA. Lt Col Reed holds a B.A. in Political Science from the Citadel, and a Masters in Public Administration from Golden Gate University.

VISIONICS DATA BASE GENERATION: AN INTEGRAL PART OF TRAINING, PLANNING AND MISSION REHEARSAL

J. Jeffrey Lombardi, Martin Marietta Flight Systems
Lt Col Edward T. Reed, 58 OG/OGU (USAF)

INTRODUCTION

Since August of 1990, a highly motivated team of government and contractor personnel has been efficiently developing large quantities of real world, photospecific data bases for combat oriented training and mission rehearsal . This government owned, contractor operated facility is now staffed with 20 data base engineers and three full time source data specialists supervised by a government intelligence and planning specialist. Experience ranges from helicopter, fixed wing, fighter, armor and sensor data base simulation programs. Much has been learned in "boldly going where no data base effort has gone before". This paper discusses the processes and standards developed as well as critical lessons learned. Now the largest government owned, contractor operated (GOCO) simulation data base generation system and library in the Department of Defense, this team has spent considerable time planning and implementing a unique approach to high quality, high accuracy rapid data base development which continues to evolve and improve to this day. Let's examines this process and learn from it's experiences, both good and bad.

LESSONS LEARNED

As the data base facility at Kirtland grew to accommodate additional tasking, the number and complexity of commonly encountered data base obstacles grew at an even quicker rate. Situations that might have normally occurred only once during the lifetime of a visual data base were now affecting the process in a multiplicative fashion. Small disjoint data bases needed combining, older data bases needed updating and moving models needed to look and act exactly the same way in every data base. Given normal production time lines, all of these issues are easily overcome using the traditional method of reformatting, retexturing and retuning. In a mission rehearsal environment it was, and still is, unacceptable as time becomes the critical factor. Compounding the situation were documentation and configuration management issues. Tracking and documenting the same files with singular differences soon accounted for nearly 30% of the man hours expended on the projects. Also, came the most difficult of lessons, that of data base conversions and source data. It was difficult to explain to a "customer" that indeed his area of interest had been constructed, however significant effort was necessary to "rebuild" the data base to fly on his visual system. To the customer, all the eloquent technical explanations boiled down to, "how long will it take". Source data became an issue early on when data base engineers began asking themselves questions such as "how tall does that fence look to you?", or "is that a wall or a hedge?". These questions brought up concerns that unqualified decisions might influence a crew to change their mission based on the virtual environment they were experiencing. The reality was painfully too obvious. After all, that's what combat oriented training and mission rehearsal is all about. Even today, data base engineers at most production sites are forced to make important decisions that are best suited for photo interpreters and image analysts. If accuracy is a critical requirement, then it's critical to have accurate source data information.

THE PROCESS

Addressing this last problem first, the acquisition of source data information to support the data base group was actually the easiest of all to solve. Consequently, a Tactical Analyst (photo interpreter), Electronic Warfare Analyst, and JARMS engineer (Jammers, Artillery, Radars, and Missile Systems) were added to the Kirtland site. It is their mission to support the data base process and enhance the simulation experience for all the users. Each member of this team has significant military experience in their associated fields. With this group of specialists formed, it was now time to address the other issues. The solution involved a two-fold approach. The first part involved the data base process itself, while the second dealt with a standardization scheme. The total solution had to accommodate rapid data base generation, conversions, connections, updates, and incorporate this newly created source data group, while the standards needed to affect every aspect of data base generation and remain flexible enough to accommodate the inherent uniqueness of each new environment.

The first task at hand was the re-evaluation of the data base process. Historically, detailed data base requirements were sketchy as air crews tried to describe in non technical terms what their mission might require from a visual data base. These discussions overlooked issues such as the source of provided latitude and longitude coordinates, the datums of these coordinates, lines of communication and obstruction data, threat information, and even the validation of the data. So who on the government side is qualified to provide this information? Thus began step one in modifying the data base process. Our local government simulator organization or OGU could ask these and other questions and if need be determine the answers *before* the data base work is ever started. The emphasis became "good source data from the start" and the adage "garbage in, garbage out" was adopted. Figure 1 illustrates how the new source data team was integrated into the data base generation process.

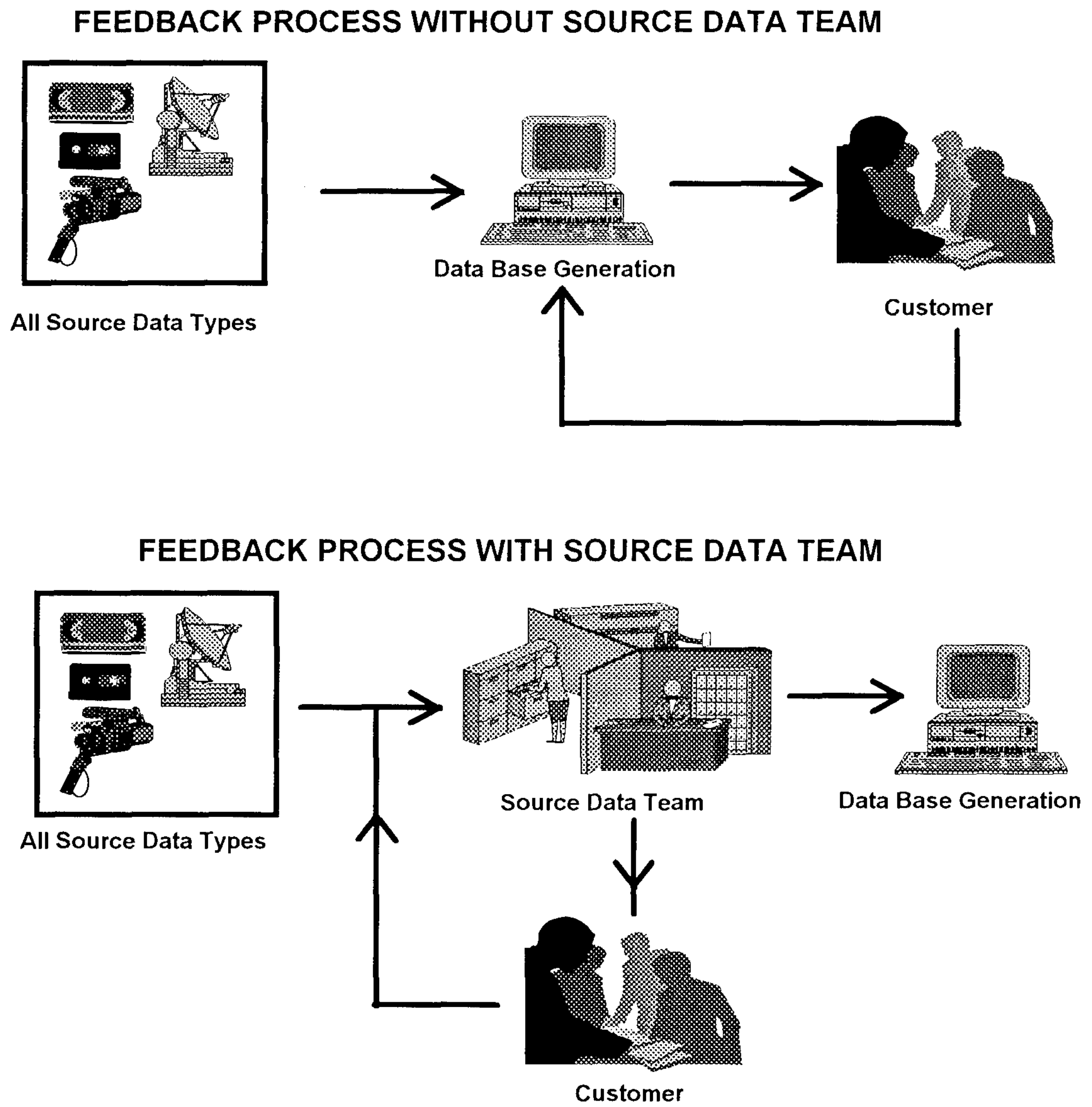


Figure 1

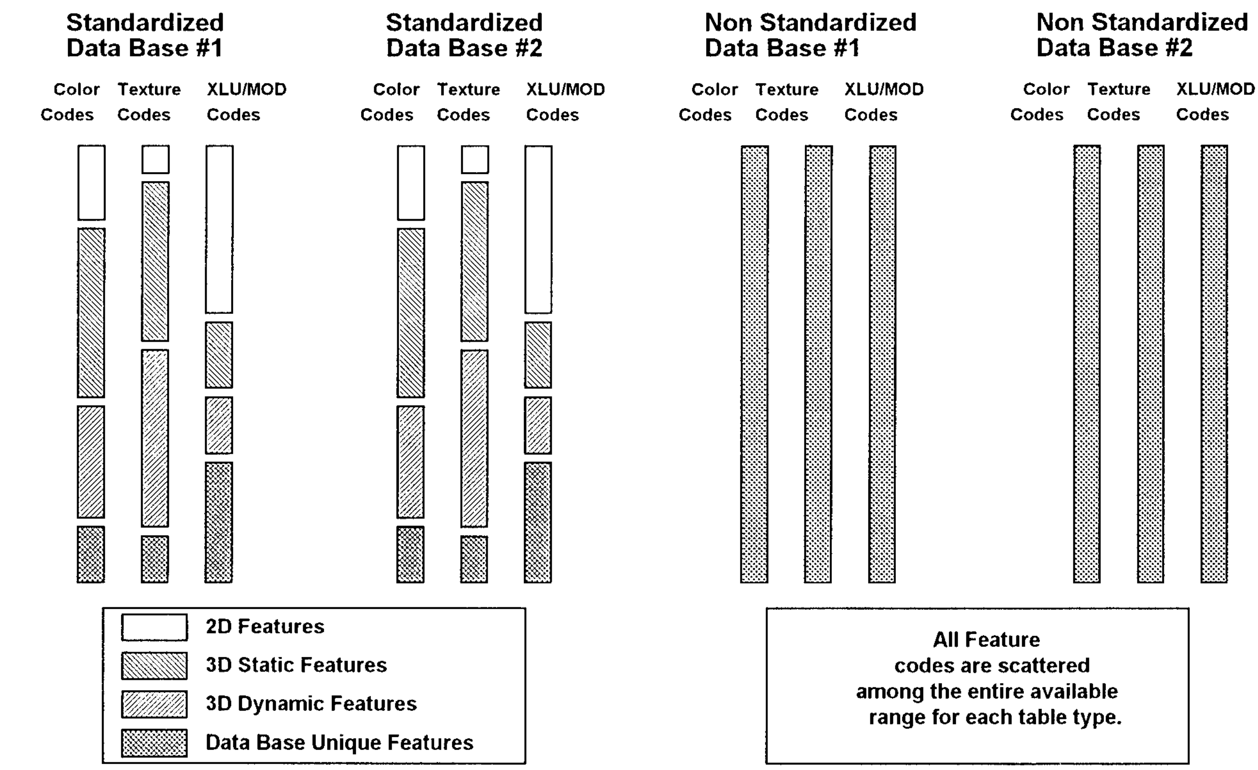
Now a procedure looked upon as ritual, our source data team is responsible for identifying coordinate datums, verifying the accuracy of these coordinates, identifying and prioritizing feature importance, and feature mensuration. They are also responsible for ordering, cataloging, and tracking every piece of source data received by the facility. Data base engineers now had a reliable, valuable, and best of all, local point of contact for source data and related information. To insure the government/contractor team has access to all types of source material of interest, the team is now equipped with state of the art mission planning workstations that allow receipt of media ranging from broadcast and VHS tape to 8mm tape, CD ROM and Laser Disk. Since it's inception in early 1991, the source data team at Kirtland has acquired, collated and processed the data needed to support both training and mission rehearsal activities. And, over the past several years, has amassed a documented data library now used by multiple "customers" in DoD for simulation support.

Drawing on the vast experience of the data base engineers, the entire team scrutinized the remaining areas of the data base process down to the smallest details involving file structures and tools used. Focusing on the tasks that required the most effort, combining data bases became the first target. The first manifestation was the realization that the construction of a data base is not a linear process. Several different tasks can be performed simultaneously and merged at some point later in the process. The key is to insure that when the merge takes place all the pieces match precisely so that no rework or duplication of work needs to take place. This merging process may be as simple as combining two files together or as complicated as combining two data bases together. Based on our experience, independently designed data bases will have to be combined into a single contiguous landmass. With additional growth planned in the future, it was originally postulated that if there were some way to insure that like features in every data base were identically attributed, combining data bases would be greatly simplified. The only time required would then be that needed to integrate the geospecific features inherent to each separate data base into the combined total. The solution was obvious. We needed to create a standardized set of global attributes and a library of three dimensional features. The global attribute set that resulted was designed to be as rich in data as the data base generation software would allow. Beginning from Defense Mapping Agency's (DMA) level 2 Digital Feature Analysis Data specification guide, it also includes all those attributes unique to visual and sensor data bases. Colors, texture references, modulation, translucency, and lod curves to name a few were all accounted for. The impetus was that regardless of what application the data base is designed for, over attribution could never be a draw back. An example of the value added can best be demonstrated using a program whose requirement may only be to produce a sensor data base. By utilizing the over attribution methodology, the customer would not only receive a data base that was correctly attributed for his sensor displays, but the out-the-window representation would also be correct. If this program were now planning to upgrade their training system to include visual displays, no time or money would be expended reworking the data base to accommodate the new requirement.

The second area of focus was data base conversions. To this end, the engineers evaluated conversions to other CompuScene image generators, back transformations to DMA format, the Standard Interchange Format (SIF) developed under project 2851, and even into formats of other image generator (IG) vendors. Because it is difficult, if not impossible, to foresee all the formats a data base might end up in, the global attribution set also includes fields that seem useless to our internal data base generation structure. Again, the more attribution the more valuable the data. Since this global attribute set has been in place, not a single man hour has been expended reworking data due to the lack of attribution. Clearly however, conversions to other image generator formats may require limited types of rework due to the intrinsic differences in IG architectures and their associated capabilities.

The standardized library of 3D features also began with DMA's level 2 DFAD specification. It was then enhanced to include vegetation models, generic buildings, animals, and several other additions. Intended to be attributed as richly as the 2D features, this library would serve as the baseline for all new data bases. During its development however, it became apparent that DMA does not support the diversity of features that might be used in a visual data base. For instance, if the desired model is a helicopter or an airplane, what FID (Feature Identification Code) do you assign? How about features such as animals, and most other transportation or military models? To solve this problem we drew upon the sensor data base experience of the group. Because sensor data bases need to support the breakout of significant "hot spots" of a target, there existed a method of assigning FID codes and then calculating the corresponding sensor colors. This method involved extending the normal set of DMA supplied FID codes that end at 999 into the 1000s range. By adopting the extended FID list and the associated Infra Red Math Model that assigns sensor color into the data base process, the three dimensional models could now be attributed equally as well as the generic DMA features. With this standardized library of 2D and 3D features, combining and converting generic data bases no longer requires the reshuffling of feature attributes or the modification of color, texture, modulation, translucency, or other tunable visual attributes. The appearance of the ocean in one data base was exactly the same as in another. This improved process meant that a data base engineer would no longer need to utilize valuable Image Generation time to validate any of the features in the standardized library. They were guaranteed to be correct. A savings of both time and money.

The next task was to migrate the standardization methodology down to the file structure level. Every reference file and look up table used during the production of a data base was evaluated for possibilities of standardization. Some files, such as lighting tables, were partitioned into aircraft, rotor wing, friendly and foe sections while others were partitioned to segregate 2D, 3D, moving models and geospecific features. The goal was to try and produce a file structure that segregated static and dynamic features as well as differentiating two dimensional from three dimensional features. It also had to be flexible enough to accommodate expansion and the unavoidable subjective tuning that all photo specific and geospecific features go through. Using a color table as an example, if a color needed modification, such as the blue of a river, we wanted to localize the change to rivers only. Hence a 3D feature that may use that same color blue on a wall or a portion of a fuselage remains unaffected. While implementing these file changes, look up tables and reference files began to appear repetitious within themselves.



Combination of these two data bases requires the merging of the data base unique ranges only.

Combination of these two data base requires significant work to separate duplicate usage of codes from each different data base.

Figure 2

Again using the color table as an example, there might be a range of shades of blue reserved for 2D features the same range repeated for 3D features and yet again for moving models. Were we actually sacrificing image generator capability for data base standardization? Well, not entirely. When an Image Generator is designed, experience dictates that little forethought is put into the analysis of actually how many colors or curves a visual data base might actually use. In reality, the numbers are usually based solely on the power of two theory, and even then are usually made as large as possible. Image Generators that tout 256 translucency curves for example normally have data bases built that use only 20-30% of those curves! Also, keep in mind that the partitions within any particular file do not need to be of equal size. A crude example might be to set aside only 5 different shades of green for two dimensional features while the three dimensional features might allocate 20 different shades. Figure 2 is a graphical representation of how a pair of standardized and non standardized data bases might have their color codes, texture codes, and translucency and modulation codes arranged. The combination of the two standardized data bases simply requires the merging of the data base unique feature ranges, while the combining of the two non standardized data bases will require significant rework to reorganize the attribution codes to make a single contiguous data base.

We now had in place a standardization scheme that encompassed all the generic and shared features that might somehow be affected during the combining of two independent data bases, or the conversion to other formats. It was now time to approach those features of a data base that made it truly unique. The geospecific features and imagery files. Because these are the features of a data base that an air crew is most interested in, the *process* by which these areas were developed became suspect for improvement. The standardization scheme already adopted up to now would only insure that the features are built technically correct. The difficult part was insuring spatial and dimensional accuracy. Oddly enough, the first step in improving the process for the development of areas of interest was education. It is difficult for a data base engineer to tell a crew member how accurate an area might be if they are unaware of the inherent inaccuracies of the sources used to create it. Thus began a rigorous education program. Gathering all the educational materials possible and again working closely with the source data team, the data base engineers learned about charts, maps, datums, imagery, and more. They became concerned about the accuracy of each and every piece of source they were using and even investigated the accuracy of the tools they were using. A more educated engineer will always result in a better end product. The results represented a significant improvement in quality and accuracy. Not only did the engineers get a tremendous education, but they were now able to help educate the multiple government customers now requesting data base products from the facility. If you want an accurate data base you must have accurate source. Crew members were sensitized to these issues and fully briefed before they entered the simulators. In some cases features in the data base were purposely discolored to indicate that unreliable data was used to create that particular area. Extending the standard methodology into this type of development resulted in standardized naming conventions for geospecific features, and the advent of new tools. When all was said and done, the development time for photo textured, geospecific areas was decreased by approximately 40%.

The last area that endured scrutiny was documentation and configuration management. Although often overlooked and their value underestimated, both of these areas were of great concern. They were taking a considerable amount of time to complete, and were at the mercy of the project leads who double as documentation authors. Some of the configuration management issues were overcome simply due to the adoption of the new standard processes. Once the "standard" data base files have been captured, only the imagery and geospecific or unique features remain. Not only did the hours to perform these tasks drop but so did the amount of removable data storage used to archive a data base. Historically, configuration management has been viewed as a cumbersome bureaucratic aristocracy, where only the chosen few were allowed to breach the security of a configuration managed library. At Kirtland we have made some marked changes to that philosophy. All the data captured by the configuration management team is available to every engineer at any time of day or night. It is stored on removable optical disks inside the data base lab itself. Maintained copacetic via standard computer security techniques, the attitude adopted was one where configuration management was viewed as necessary, but should not be seen as inaccessible. Documentation was another relatively easy area to implement a standard philosophy. Document templates for each required document were created. This method requires engineers to merely cut and paste relative information while keeping the standard verbiage intact. This approach also proves handy for the customer as they have become accustomed to the standard documentation format for each data base. Still a time consuming chore, efforts are under way to automate the documentation process to further reduce what will always be viewed by an engineer as drudgery. This standard of configuration management will allow easier interchange of data bases between DoD customers using the Mil-Standards established by the joint service Project 2851

IMPLEMENTATION

At last the process had been streamlined and a comprehensive standard agreed upon. But how were we going to ensure it was used? Accessibility and consistency were now the focus. Just as people are disappointed by a restaurant whose meals taste different every time, if our standards were not consistent and accessible they would inevitably fail in reaching their ultimate goal. The standard data base, as is it commonly referred to, is configured on the data base network just as every other data base is. It is also captured on removable media by configuration management. To keep the standard process evolving, weekly unit meetings are used as a forum for engineers to present ideas or modifications. The files on the system are left unprotected to allow engineers to use this data as a testing ground before any formal presentations are made. When additions or modifications to the standard need to be formalized, all the engineers are queried for their evaluation of the change. The final approval rests on the core group of lead engineers. To insure that new engineers were able to learn these standards and apply them correctly, a complete set of documentation was maintained throughout the entire development cycle of the processes and procedures. Now referred to as the "standard document" it serves as a bible to help guide engineers through visual data base development. In addition, the government produced a five hour course on data base generation and a text on source data for simulation data bases.

To help insure the implementation of the new processes, several new tools were designed and built by the engineers at Kirtland. The concept was to eliminate as much user induced error from the process as possible. Through painful experience something as insignificant as a typographical error can wreak havoc during the generation of a data base costing hours or even days of progress. The first step in reducing error was to reduce that actual work load required to complete a specific task. Take building a model for instance. The implementation of all the new standards and processes meant that the creation of something as simple as a three dimensional box became a complex task. Paging through documentation searching for the correct ranges in attribute files, look up tables, lighting files and numerous other files just meant more opportunity for error to creep in. To address model building in particular, a task that takes up about 50% of data base development, our engineers generated their own primitive model tool. Customized for our specific requirements, it incorporates all of the a-fore-mentioned standards and combines them into an easy to use graphical user interface, while maintaining all of the interdependencies encountered in feature attribution. The selection of a FID code for instance automatically reduces to the modeler's options for surface material codes and other parameters to only those values identified as valid per the adopted standard. Originally targeted for inexperienced data base engineers, we found that sometimes too much flexibility was a hindrance. None-the-less this home brewed modeler has enjoyed tremendous success and is now used by everyone in the group to create nearly 80-90% of the generic structures in a data base.

THE RESULTS

To validate the end result, detailed metrics for nearly every individual task were tracked both before and after the implementation of these the new processes and standards. Trends and bottlenecks were identified and corrective action taken if needed. The group had reduced the original data base production time by more than 50%, with continuing reductions in the new future! The accumulation and tracking of all the data even lead to the development of a data base estimating tool that has proven to be incredibly accurate. Even as this paper is written, new standards are being developed to accommodate the new generation of image generators. In the words of our own data base engineers, a standard is not only a way of doing things it's an attitude. Things get done right the first time. As another engineer put it, if you're not going to take the time to do it right the first time when will you take the time to correct it? Of course the final and most important tests are performed everyday; by our customer. Since the implementation of these standard processes the data base group at Kirtland has had one data base pass through a fully integrated ATP without a single test discrepancy, and most other data bases have discrepancy numbers down in the 10s. Our confidence in this process has been further confirmed by spatial integrity tests verified against real world known information and conducted by an independent government agency. Their evaluation proved the old adage of "garbage in, garbage out" and validated our accuracies when provided quality source data.

The other key aspect of the finished product is correlation. If a training system has an environment that includes out the window, radar, FLIR, NVG, navigational aids, and Electronic Warfare data bases, it is essential that they are 100% correlated. The easiest way to insure correlation is to build all of these data bases from a single source file. A source file so rich in attribution, so consistent, that it carries within its file structure all the data required to genesis these other data bases. At Kirtland that is exactly what we do. As a member of the joint government/contractor team at Kirtland , we believe we maintain some the highest standards in the industry, and have four years of training and mission rehearsal data base experience to prove it. We hope sharing our experiences will benefit other DoD customers and contractors now and in the future.

THE FUTURE

The future is bright for the continued expansion of simulation data base efforts at Kirtland. In 1994, the Air Force established the Simulator Data Base Facility or SDBF at Kirtland. The SDBF is the operational implementation of the joint service Project 2851 and the culmination of twelve years of research and development which have resulted in the two key military standards for the economic development and interchange of simulator data bases. These Mil-Stds, 1820 and 1821 will greatly increase data base productivity and availability. The SDBF will library and enhance data bases and models in these standards as well as contractor formatted run time data bases and fully prepared geo-referenced imagery. DoD customers can order standard or enhanced data bases for a small fee for service. The SDBF is expected to save millions of dollars in its first two years of operation by eliminating the current duplication of efforts in data base and model development. The SDBF further establishes Kirtland as a center of excellence for DoD simulation data base development and modification. We encourage all DoD customers to take advantage of the SDBF to allow concentration of program dollars to customize already developed data bases instead of recreating them.

Statistical Certification of Terrain Databases

Dr. Guy A. Schiavone, Russell S. Nelson and Brian Goldiez
Institute for Simulation and Training
3280 Progress Drive
Orlando, FL 32826

Abstract

Consistency in terrain representations between run-time databases is a prerequisite for interoperability in Distributed Interactive Simulation (DIS). It has been suggested in previous research that one hundred percent alignment of databases will never occur in a simulation that utilizes distributed geometric databases. However, statistical certification of terrain database elevations offers a means of ensuring the degree of consistency necessary for interoperability. In this paper we define a statistical metric for terrain database certification. Starting with a review of the existing work on quantitative terrain database metrics, we examine a basis for specification and statistical certification of terrain elevation data. Using classical acceptance sampling, hypothesis testing will be introduced as a method by which a terrain database (TDB) is certified. A method for determining the critical error value for the desired accuracy proportion and consumers risk (Type II error) will be discussed. From these results the producers risk associated with the test is evaluated for several different accuracy proportions. Using data collected at the 1992 I/ITSEC as a basis for comparison, the utility of acceptance sampling is demonstrated using data collected at the 1994 I/ITSEC. A distinction is drawn between tests designed for TDB certification and tests with inherent diagnostic capability. As an example of the latter, the use of the cross-correlation metric is introduced for the purpose of detecting linear shifts between the terrain skins of a baseline database and a trial database. Using a portion of the Hunter-Liggett high definition area, an example of linear shift detection is provided for the case of a shift by an integer number of samples.

About the Authors

Dr. Guy A. Schiavone is a Visual Systems Scientist at the Institute for Simulation and Training. He holds a Bachelor of Engineering from Youngstown State University, and the Ph. D. in Engineering Science from Dartmouth College, Thayer School of Engineering. His current interests include spatial error in terrain databases, 2-D signal processing, image processing, scattering from random surfaces, and propagation through random media.

Russell S. Nelson is an Assistant Engineer at the Institute for Simulation and Training. He holds a Bachelor of Science in Electrical Engineering and a Master of Science in Electrical Engineering, both from the University of Central Florida. Currently, Mr. Nelson is a member of the Visual Systems Laboratory R & D of Terrain Databases for DIS project. Prior to joining the VSL, Mr. Nelson was a Graduate Research Assistant with the Distributed Interactive Simulation Lab at IST.

Brian Goldiez is the Director of Research and Development at the Institute for Simulation and Training. Mr. Goldiez's professional interests are in aerodynamic modeling, visual systems, systems design, and testing. Mr. Goldiez directed IST's efforts in the first large scale design, test, and demonstration of Distributed Interactive Simulation at I/ITSEC 1992. He has been in simulator research and development for over 15 years as an employee of the US DoD, industry, and academia. Mr. Goldiez has Bachelor of Science in Aerospace Engineering and a Master of Science in Computer Engineering.

Statistical Certification of Terrain Databases

Dr. Guy A. Schiavone, Russell S. Nelson, and Brian Goldiez
Institute for Simulation and Training
3280 Progress Drive
Orlando, FL 32826

Introduction

In recent years, the effectiveness and relative low cost of applications utilizing Distributed Interactive Simulation (DIS) has made the development of DIS a focus of the US military for the purpose of training and other applications requiring real-time interactive simulation. DIS is defined as a time and space coherent synthetic representation of world environments designed for linking the interactive, free play activities of people in operational exercises [1]. Each simulator on a DIS network maintains its own representation of the world. While current technology has provided visual systems with the capability of displaying high fidelity representations of a given synthetic environment, the failure to define and properly certify an agreed-upon synthetic environment before the start of a simulation exercise can lead to significant inconsistencies between world views of individual entities. Also, while members of the simulator industry compete to simulate operational systems at minimum cost to users, there are often times when proprietary "black-box" implementations lead to interoperability problems between networked simulators. It is well known that a consistent playing field between all networked simulators is essential to a successful training mission or evaluation of a new weapon system. As recently noted by Woodard [2], a fundamental first step in addressing this problem is the establishment of a common database format and content. In order to ensure that the content remains unaltered in the transformation between source database and runtime database, it follows that the content of the individual run-time databases should be tested and certified as a necessary step to guarantee a successful simulation exercise.

There is an industry consensus that the most common sources of spatial error between virtual environment representations in networked simulators include inaccurate coordinate transforms, TDB preprocessing by graphics systems, differences between rendering algorithms, and inconsistent source TDBs [3]. As an example, data and analysis recently presented by Economy, et. al. showed inaccurate coordinate transformations to be a leading source of positional error between simulators in end to end system tests [4]. Although progress is being made to improve the quality and consistency of rendered images through improvements in hardware technology, it has been suggested that resolution of interoperability problems by hardware improvements alone is not in the foreseeable future. However, in simulation environments consistency checks can be applied between TDBs as well as between displays. Thus, there exists an avenue on which the problem may be approached, and that is by way of certification testing of runtime terrain databases. Although, in this paper the authors concentrate on terrain skin only, the environment includes space, atmosphere, earth and sea; features and attributes as well as elevations. Ultimately, spatial coherence metrics for all features of the synthetic environment must be developed. Goldiez, et. al. have mentioned that a spatially coherent environment is an essential element to achieving non-biased simulator interaction [5]. Furthermore, any interaction that takes place in an environment that is spatially incoherent would be accidental and probably meaningless. It is understood that one hundred percent coherence between runtime TDBs is not currently feasible due to performance differences and other causes, and this suggests that applications-based acceptance criteria for runtime TDBs must be defined.

Although much attention has been recently given to the issue of interconsistency between terrain databases in the DIS world, terrain database "correlation" has been recognized as a problem in the real-time simulation community since at least 1977 [6] . Since that time many qualitative evaluations and discussions of the problem have appeared in the literature (for example, [7-12]). Other references can be found in a survey conducted by Zvolanek and Dillard [13]. Unfortunately, proposals for attacking the problem on a sound quantitative footing have been infrequent. Zvolanek and Dillard [14] evaluate terrain elevation "correlation" by calculating the statistical mean, standard deviation, and range of the elevation differences. Feature "correlation" is defined as the percentage of misclassified pixels. Dunn-Roberts et. al. propose a line-of-sight (LOS) intervisibility metric to measure differences in intervisibility between two TDBs [15]. A LOS comparison metric was also used by Fatale et. al. [16] in a study comparing DTED levels 1 and 2. Ellis [17] recommends measuring off-line elevation errors by the statistical mean and the 90% or 99% maximum error, per unit of standard roughness.

Even though the quantitative methods outlined above represent the current state of the art in terrain database spatial error metrics, they all suffer from various shortcomings. There exists no criteria or guidelines to determine an acceptable level of error for a given application, or how to use the results of the tests. The statistical metrics mentioned above do not allow for control or estimate of producers and consumers risk. The simple statistical measures yield no information on error locality, beyond human-in-the -loop visualization of difference maps. The LOS methods may require a large number of calculations since, in order to obtain a unique error mapping, intervisibility must be calculated from every point in each TDB to every other point in the TDB. None of the metrics mentioned thus far are diagnostic in the sense that they are able to detect shifts, rotations, warps, or other spatial or temporal characteristics of the error. Moreover, there has been limited attention given to identifying the source of the error between a source TDB and a subject TDB. A preliminary investigation of some TDB metrics that overcome some of these shortcoming was undertaken by Kilby et. al. [18]. Currently, IST is involved with a STRICOM funded project to define and quantify interoperability in the DIS paradigm. It is the intention of this paper to present a solid mathematical approach to quantifying differences between TDBs. Acceptance sampling techniques will be applied to the elevation differences between two TDBs. Thus, an accuracy proportion with an associated confidence level can be determined and used to establish the degree of error of the subject TDB. Examples of acceptance sampling will be given using data collected at the 1993 I/ITSEC. Moreover, the use of the cross-correlation for the purpose of linear shift detection between TDBs will be investigated.

ACCEPTANCE SAMPLING THEORY

Acceptance sampling is the branch of statistical quality control that is concerned with calculating the risks associated with accepting or rejecting product lots based on information provided by a sample of the lot. Originally developed for industrial purposes, acceptance sampling was first used for map accuracy certification by Ginevan in 1979 [19]. As a background to testing for TDB elevation accuracy using acceptance sampling, we begin by examining a sample of terrain skin sample points and calculating the elevation differences, Δz_i, between the source database and the runtime database under test. The sample elevation differences are then compared to a given maximum elevation error criteria Δz_0. For example, denoting $\Delta z_1 \ldots \Delta z_N$ as our elevation samples, and choosing $\Delta z_0 = 0.5$ meters as our maximum allowable elevation error, we conduct a Bernoulli trial for each sample elevation difference Δz_i, $i = 1, \ldots, N$, where the trial is counted as a success if $\Delta z_i < \Delta z_0$, and otherwise is counted as a failure. The N Bernoulli trials form a binomial probability distribution, where the binomial probability density function is given by

$$f(Y;N,Q) = \frac{N!}{Y!(N-Y)!} Q^{N-Y}(1-Q)^Y \qquad (1)$$

where Q is the accuracy proportion, N is the total number of elevation difference samples, and Y is the number of failures.

A hypothesis testing criteria will be used in this statistical approach in which the null hypothesis H_0 states that the actual accuracy proportion of the TDB Q_a under test is less than the desired accuracy proportion Q. The possible outcomes of such a hypothesis test are listed below in Table 1.

Hypothesis Ho: Q > Qa Test Conclusions	Ho is TRUE	Ho is FALSE
Do not reject Ho (Do not certify TDB)	Correct	Type II Error
Reject Ho (Certify the TDB)	Type I Error	Correct

Table 1 Hypothesis Test Outcomes

As Table 1 shows, the test yields correct results either if H_0 is true and the test rejects the database or if H_0 is false and the test certifies the database. Type I error occurs if the test certifies an unacceptable database. This is known as the consumers risk, which occurs with a probability β. Alternately, Type II error occurs when a good database is rejected by the test. This second type of error is known as the producers risk, and occurs with a probability α.

To apply acceptance sampling for TDB accuracy certification β and Q_L are chosen, where Q_L is a low accuracy proportion that will be rejected with a probability $(1-\beta)$. Note that $Q_L = Q$ in Table 1. After determining an appropriate sample size N, find the largest value X such that

$$\beta \geq \sum_{Y=0}^{X} \frac{N!}{Y!(N-Y)!} Q_L^{N-Y}(1-Q_L)^Y \qquad (2)$$

For a given N and β, the resulting value of X is known as the critical value. By ordering our elevation difference samples Δz in decreasing order, and counting down to the X^{th} sample, we determine our maximum error criterion $\Delta z(X)$ for which we may make the statement that the trial TDB agrees with the source TDB to within an error of $\Delta z(X)$ with an accuracy proportion of Q_L and a confidence of $(1-\beta)$. For example, choosing N=2000, Q_L=0.95 and β=0.05, we find that X=83. In this case we will count down to the 83rd largest error $\Delta z(83)$. If we find that, say, $\Delta z(83)$=0.5 meters, then we may say with 95% confidence that 95% of the trial TDB agrees with the source TDB to within 0.5 meters.

Once the critical value X has been determined, the producers risk a can be determined for various high accuracy proportions Q_H from the relationship

$$\alpha = \sum_{Y=X+1}^{N} \frac{N!}{Y!(N-Y)!} Q_H^{N-Y}(1-Q_H)^Y \qquad (3)$$

Terrain skin elevation tests conducted by IST at the 1993 I/ITSEC utilized a sample size of 2000. To maintain continuity with last years test, participants in the 1994 I/ITSEC demonstrations were also asked to supply a sample of N=2000 elevations points from their run-time databases. Using Eqn. 2, the critical values associated with sample sizes ranging from 1897 to 2093 were calculated for a nominal β=0.05 and a low accuracy proportion Q_L=0.95, and are shown in Table 2, below.

Sample Size N	Critical Value X	Consumers Risk β
1897	79	0.0500
1919	80	0.0498
1941	81	0.0498
1963	82	0.0496
1984	83	0.0500
2006	84	0.0499
2028	85	0.0497
2050	86	0.0497
2071	87	0.0500
2093	88	0.0499

Table 2. Optimum sample sizes N for given critical values of X and a nominal β=0.05, with the low accuracy proportion Q_L set to Q_L=0.95

Using Eqn. 3 and the sample sizes and critical values shown in Table 2, we may then calculate our producers risk, α, for some relevant values of Q_H, as shown in Table 3.

Sample Size N	Critical Value X	α for QH = 0.925	α for QH = 0.950	α for QH = 0.975
1897	79	1.0000	0.9500	0.0000
1919	80	1.0000	0.9502	0.0000
1941	81	1.0000	0.9502	0.0000
1963	82	1.0000	0.9504	0.0000
1984	83	1.0000	0.9500	0.0000
2006	84	1.0000	0.9501	0.0000
2028	85	1.0000	0.9503	0.0000
2050	86	1.0000	0.9503	0.0000
2071	87	1.0000	0.9500	0.0000
2093	88	1.0000	0.9501	0.0000

Table 3. Values of the producers risk α for various high accuracy proportions Q_H, for the values of β and Q_L used in Table 2.

We note in Table 3 that for these relatively large samples the range of significant producers risk about $Q_L=Q_H$ is very small. In the above table we see that for Q_H = 0.925 < Q_L=0.950 we have α=1.0000, which indicates the near certainty that a TDB with accuracy proportion of 92.5% will be rejected. On the other side, we see that for Q_H=0.975 > QL = 0.950, we get α = 0.0000, which tells us that there is virtually no chance of rejecting a TDB with an accuracy proportion of 97.5%. Finally, we note that when $Q_H=Q_L$, we have the case where $\alpha=1-\beta$.

Since the binomial distribution is discrete, there exists several values of N for each critical value X. Based on our test procedure of the previous year, we have chosen our sample size as N=2000, which falls between N=1984 and N=2006 in Table 2. Our critical value is therefore 83. The results of our application of the acceptance sampling theory to samples obtained from participants in the 1993 I/ITSEC will be detailed in a later section.

At the Interservice and Industry Training, Simulation, and Education Conferences (I/ITSEC)

As a part of the preparation for the 1993 I/ITSEC DIS demonstration, IST generated and distributed 2000 uniform random sample points within a geographic area of Fort Hunter Liggett, CA. This area was designated the "high detail area" because it was the only area in the data base where ground interaction was allowed. The high detail area was 10km X 30km. The latitude and longitude of the sample points were chosen at random using a bivariate uniform random distribution. Sampling of the 1992 Hunter-Liggett high detailed area was done on a grid with a minimal spacing of one arc second between samples. These points were chosen within boundaries that are one arc second toward the inside of the boundaries to avoid the effects of feathering at the boundaries. As a result of the post 1992 I/ITSEC TDB testing, a discussion on an acceptable TDB metrics followed at the 1993 I/ITSEC planning meetings. It was requested by I/ITSEC planning meeting members that the resolution of the grid from which the random points were chosen have a spacing of 0.01 arc seconds, which results in sampling with a minimum possible spacing of 0.3 meter (approximately one foot).

I/ITSEC 92

As a result of the data gathered from the I/ITSEC '92 demonstration, development of an analysis tool that allows a database engineer to locate regions of spatial error while building a database was indicated. If the differences in elevations between two databases are recursively examined and adjusted while building a database then the error in elevation can be minimized (see Fig. 1).

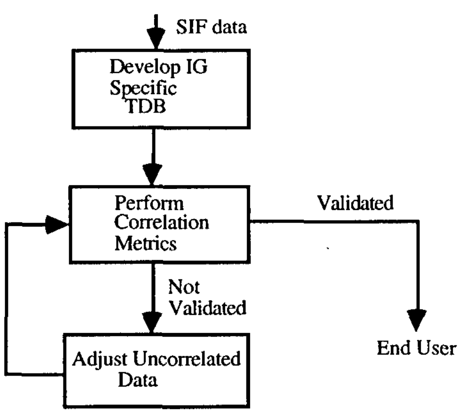


Fig. 1. Reducing Error in TDBs

The area of interest for the terrain correlation study was the high detailed inset area of the Hunter-Liggett database agreed upon for I/ITSEC 92. The area consisted of a patch of land that was bounded by a 30km easting and a 10km northing.

Based upon initial recommendations from the I/ITSEC 93 planning meeting attendees, a maximum desirable variance between a subject database and PRC's database was 0.5 meters. 1992 I/ITSEC data was analyzed to determine the suitability of 0.5m. However, after reviewing the data it became evident that a 0.5 meter error threshold would not allow anyone to participate according to the hypothesis test requirements set for 95% confidence and 95% probability for success. A filtering mechanism was used to find the number and location of the coordinates that exceeded this half-meter threshold. As seen in Table 4 very few of the participants met the 95 percent success rate at 1.25 meters.

The wide variation in 1992 data drove IST to recommend using the mean and standard deviation as criteria for 1993. We did not know at the time that in 1992 participants used gridded data as source material, when polygonal data would have been more appropriate. The polygonized SIF data was used as the standard database for I/ITSEC in 1993. Statistical analysis on the discrepancies between the subject and datum (PRC P-2851) databases showed a mean and standard deviation of the errors, as shown in Table 5.

Company	0.5m	0.75m	1.0m	1.25m
A	1719	1546	1380	1265
B	1012	641	473	340
C	1878	1815	1743	1688
D	811	422	0	0

Table 4 Failure Rate at Various Threshold Levels I/ITSEC '92 Results

Company	Mean (m)	Standard Deviation (m)
A	456	.286
B	.451	.961
C	.632	5.67
D	1.322	14.01

Table 5 Statistics for I/ITSEC '92 Databases

The data that was returned by participating organizations in the 1992 I/ITSEC revealed that the largest errors were found in geographical regions with a large variance of elevation (mountainous regions). The scatter plot of one participant indicating the points filtered from the 2000 random points that exceeded the tolerance level, defining an error, is plotted in Fig. 3. This figure represents the tolerance threshold being set at 10m. Results from other organizations reflected similar error responses.

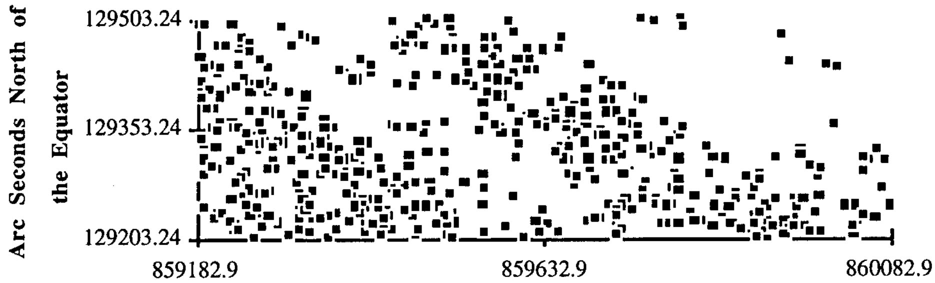


Fig. 3. Tolerance = 10m

I/ITSEC 93

Again, the area of interest for the terrain correlation study for I/ITSEC 93 was the high detailed inset area of the Hunter-Liggett (HL) database agreed upon for I/ITSEC 92. However the boundaries of the database changed from the previous year. The new boundaries were shifted north by 2km from I/ITSEC 92. Again, the area consisted of a patch of land that was bounded by a 30km easting and a 10km northing. The area for I/ITSEC 93 correlation study can be seen on a UTM map projection in Fig. 4. The distribution of the points for I/ITSEC 93 was uniform just as the sample points in 1992. In Fig. 5 notice that the range of values for elevation differences has been reduced drastically from the data collected from I/ITSEC 92, as seen in Tables (4)(5)(6). Thus, most '93 databases contained mean elevation errors on the order of a few centimeters, and errors located in the mountainous regions were greatly reduced as compared to the previous year. After reviewing the results one can note that the sample distributions for I/ITSEC 93 participants indicates that the TDBs were built with more precision than in the previous year. Fig. 5 shows the distribution of the elevation differences between the original SIF 3D

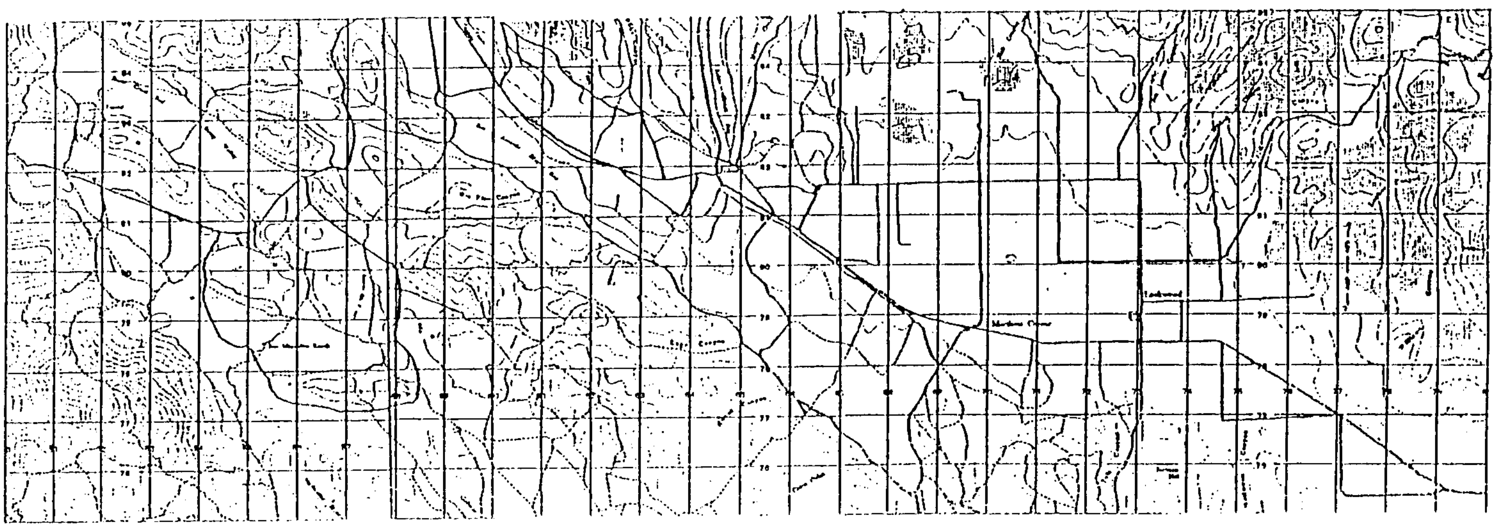


Fig. 4. Hunter-Liggett UTM Map

polygonal terrain database and the subject terrain databases. Table 6 shows the mean elevation differences, elevation difference standard deviations and the critical values. The critical value represents the 83rd largest value after finding the descending rank order of the magnitude of the elevation differences. Referring back to our previous discussion on acceptance sampling (see Table 3), the 83rd value represents the maximum number of errors allowed in a sample size of 2000 for 95% confidence that 95% of the sample points are within some tolerance (in meters) of the source terrain database. For each database the tolerance level will be different. For example, the critical value for company I was 0.162333m and the critical value for company D was 0.000163m. This means that the accuracy proportion for company I is much smaller than that of company D at any given elevation difference, while assuming a 95% level of confidence. Another interesting result of this experiment shows that although a TDB might have a negligible mean elevation difference the critical value could remain relatively large. For example, let's consider the results for company F. The mean elevation difference for company F was 0.000097m, where as the corresponding critical value was 0.029338m. Notice that the standard deviation of the elevation differences for company F is relatively large also. This indicates that there are outliers present in the elevation difference distribution.. This is shown in Fig. 5f and Table 6. In one respect, the histograms in Fig. 5 show a high correlated database with a negligible elevation shift. However, the corresponding data in Table 6 indicates that the mean elevation difference for company B was -0.019578, while the histogram in Fig. 5b appears to be shifted to the right of zero. This is caused by outliers that are present but are not within the range of the graph, which probably are indicative of anomalies in the TDB construction such as sliver polygons. Let's now compare elevation difference distributions for companies C and I. Companies C and I have relatively close statistical values (as seen in Table 6), however, the histograms in Fig. 5 show that the error in the company C database is less central than that of company I. Company I could shift the elevation of their entire database by their average Δz to correct the error between their database and the source database. However, since company C's elevation differences are not as central, the correction procedure is not as simple.

Company	Mean Delta-Z (meters)	Std Deviation (meters)	Critical Value (meters)
A	-0.00079	0.029533	0.011836
B	-0.019578	1.382213	0.069666
C	0.022796	0.603328	0.015828
D	-0.000002	0.000212	0.000163
E	-0.000065	0.015944	0.007024
F	0.000097	0.073186	0.029338
G	0.000000	0.004332	0.001501
H	-0.000090	0.009548	0.005330
I	0.487752	0.361715	0.162333

Table 6. Statistics for I/ITSEC 93 Databases

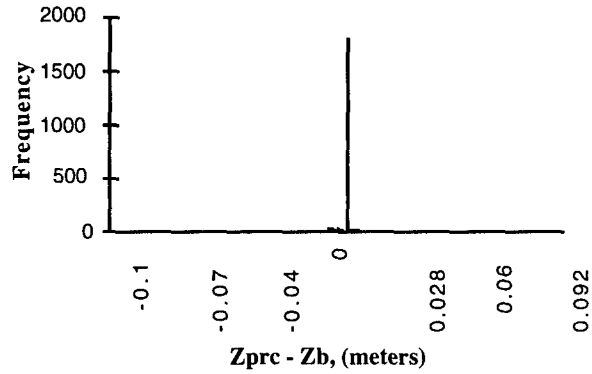


Fig. 5b

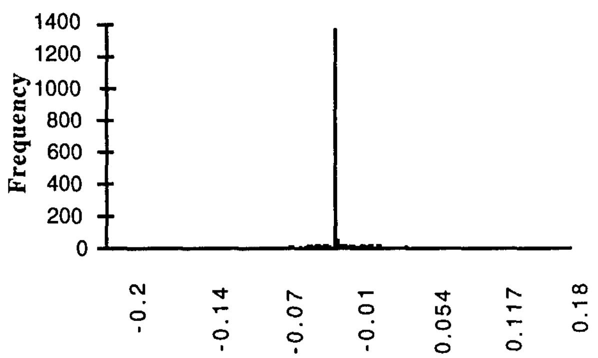


Fig. 5a

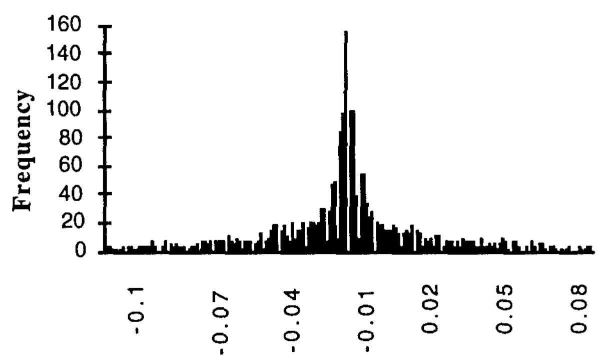


Fig. 5c

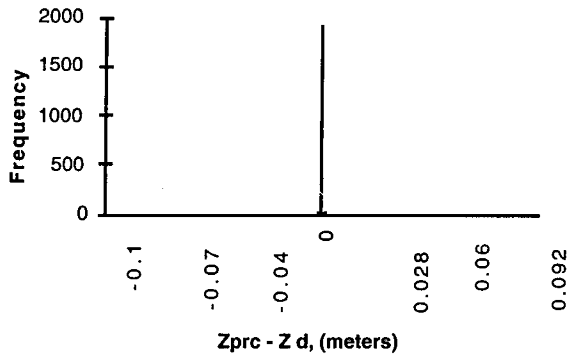


Fig. 5d

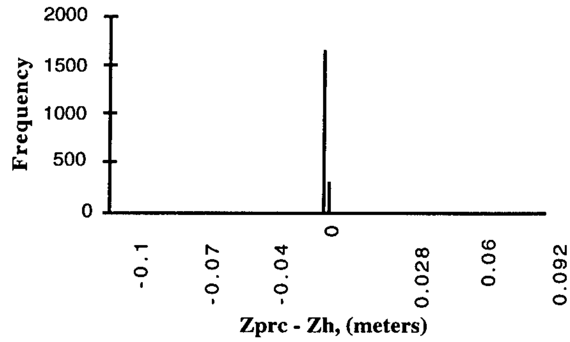


Fig. 5h

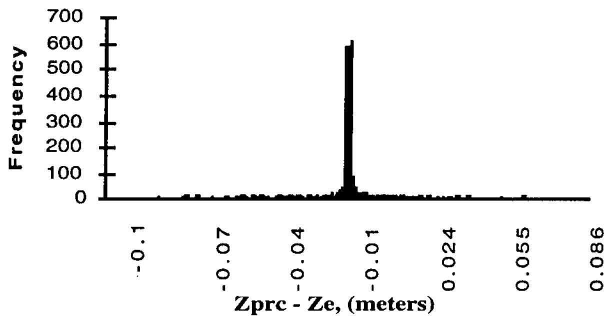


Fig. 5e

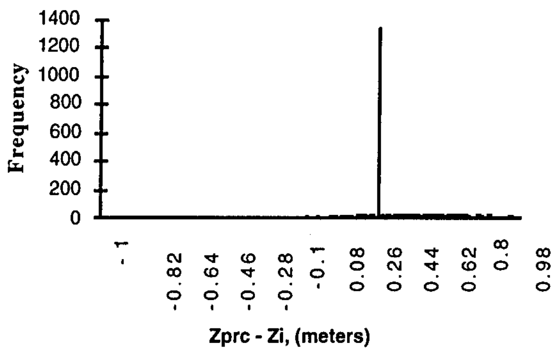


Fig. 5i

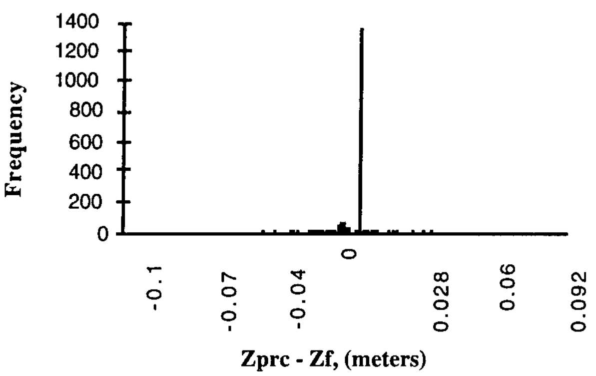


Fig. 5f

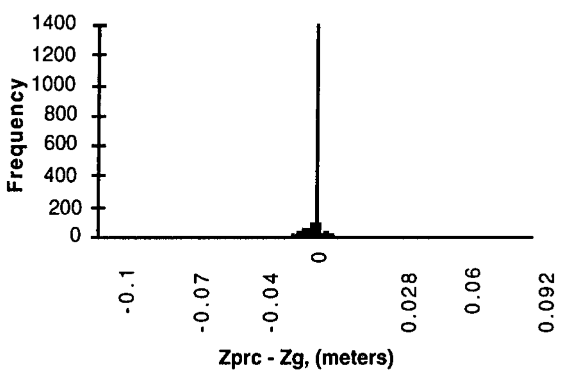


Fig. 5g

CROSS-CORRELATION TESTING

As opposed to the probabilistic statements as to the TDB spatial error made possible by acceptance sampling theory, a diagnostic metric should not only provide a measure of spatial error but should also extract information as to the type of error. Discrepancies specifically mentioned by Kilby [18] are shifts, skews, warping and resampling. Economy [3] observes a linear shift due to a suspect coordinate transformation. The ability to detect, say, the magnitude and direction of a simple linear shift in coordinates may allow one to easily determine the source of the error.

Our initial approach in developing run-time CIG-specific database correlation diagnostics is to consider the full cross-correlation on the gridded elevation data. A requirement of this approach is that a symmetric and uniform grid of elevation values must be extracted from the run-time database. Given G, a K x L set of baseline data, and H, an M x N set of trial data (with M < K and N < L), the normalized correlation of lag (k,ℓ) between G and H is

$$R_{k,\ell}(g,h) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(g_{i+k,j+\ell}-\bar{g}\right)\left(h_{i,j}-\bar{h}\right)}{\sqrt{\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\left(g_{i+k,j+\ell}-\bar{g}\right)^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\left(h_{i,j}-\bar{h}\right)^2\right)}} \tag{4}$$

R will range between -1 and 1, with R=1 describing perfect correlation, R=0 describing a complete lack of correlation, and R=-1 describing perfect anticorrelation. The initial approach is to compute R for every possible lag (k,l). The method could possibly be refined by investigating methods of determining the path that leads to the global maximum, without having to compute every possible lag. This form of the correlation will be most useful in determining linear shifts in the xy-plane. Other forms can be developed to measure other types of discrepancies. We expect this method to succeed for any reasonable data sets, although certain special cases can be constructed where, in the absence of special provisions, the method would fail, such as cases where in the windowed region the terrain elevations are doubly periodic or periodic in one dimension and constant in another.

An example of the utility of the cross-correlation metric comes from a preliminary test conducted at IST. Fig. 6a shows a portion of the terrain skin from the 1993 I/ITSEC high-detail source database, slightly upsampled at every 100 meters. The terrain extends 6.4 kilometers north and east from the southwest corner of the Hunter-Liggett highly detailed area. In the test, the data used as the baseline data was the first 60 by 60 samples, while the trial data used was also a 60 by 60 sample of the terrain skin, but shifted by 400 meters (4 samples) both to the north and to the east. The cross-correlation of these two data sets is shown in Fig. 6b. The maximum value, as given by Eqn. 4, is found as $R_{5,5} = 1.0$. Thus, the correlation returns the exact linear shift for this case involving a shift by an integer number of sample intervals.

Fig. 6 a) Southwest corner of Hunter-Liggett high-detail area. b) Cross-correlation of two different sample sets from Fig. 6a, with the second set shifted both to the north and east by 400 meters.

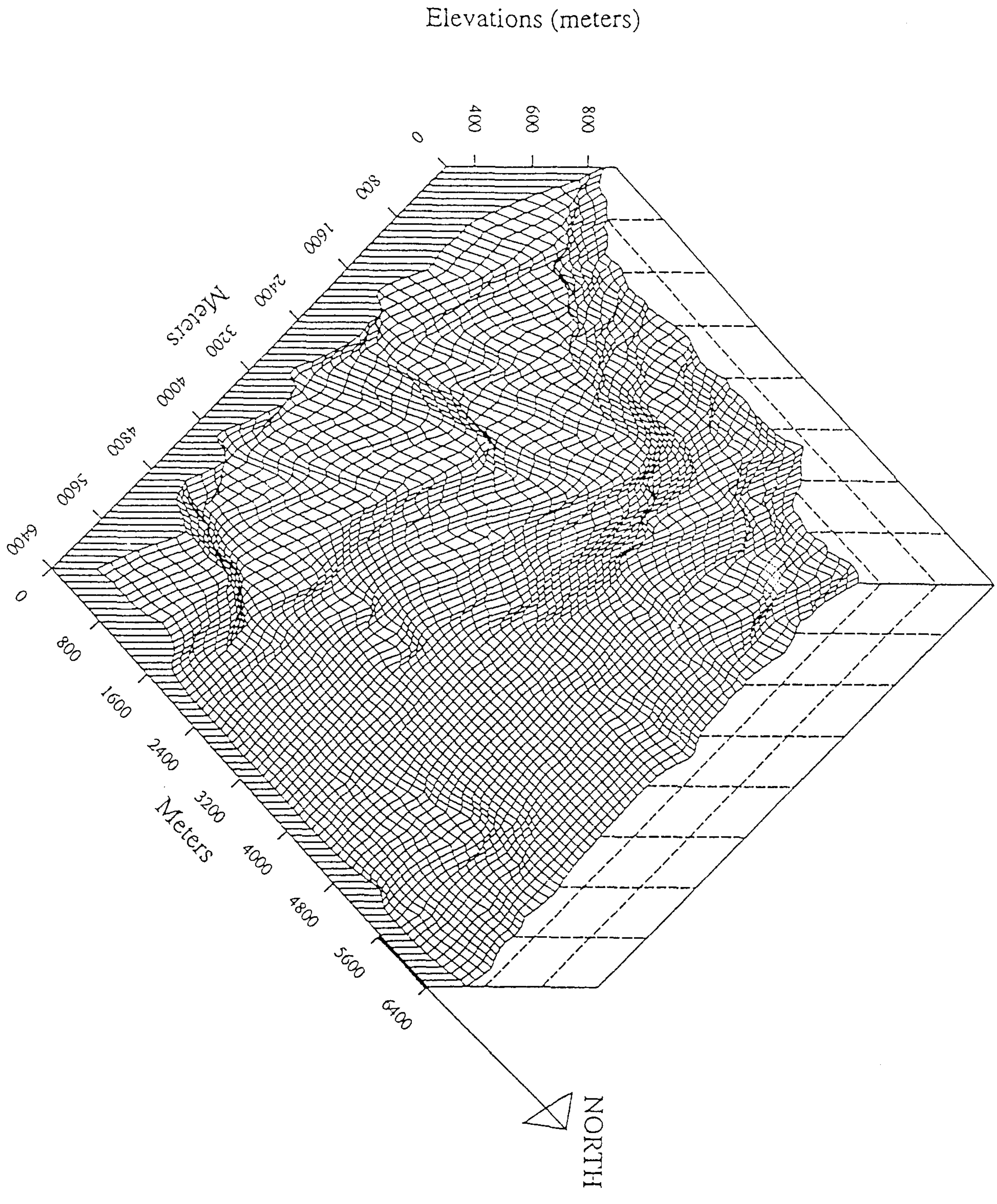

Elevations (meters)
400
600
800
0
800
1600
2400
3200
4000
4800
5600
6400
Meters
0
800
1600
2400
3200
4000
4800
5600
6400
Meters
NORTH

Figure 6a

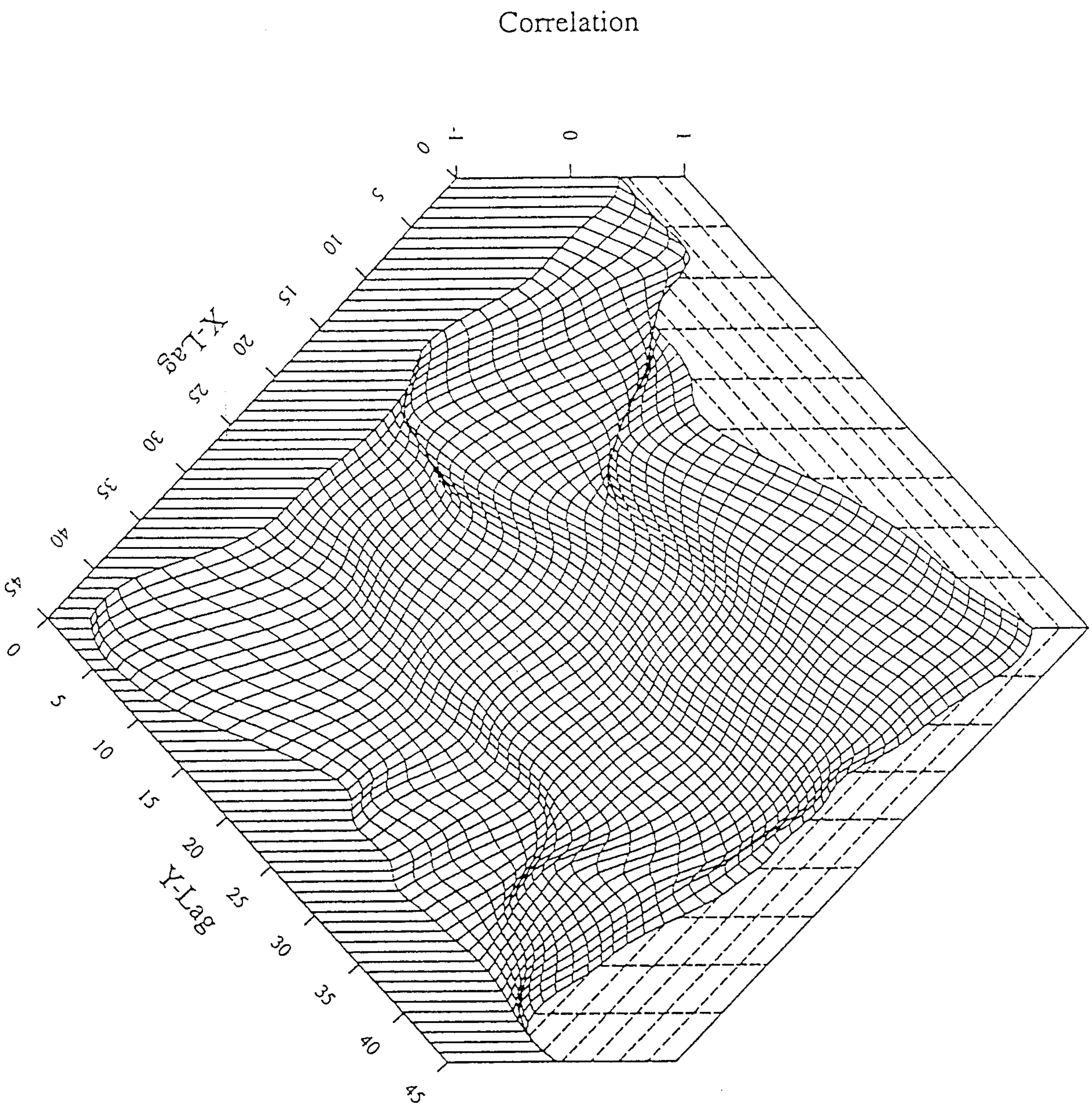

Correlation
-1
0
1
X-Lag
0
5
10
15
20
25
30
35
40
45
Y-Lag
0
5
10
15
20
25
30
35
40
45

Figure 6b

REFERENCES

[1] Standard for Distributed Interactive Simulation -- Applications Protocols, Version 2.0, Fourth Draft, Institute for Simulation and Training Document IST-CR-93-40, February, 1994.

[2] Pamela S. Woodard, "Requirements for Interoperable Environments", Proceeding of the IMAGE VII Conference, pp. 258-271, Tucson, Arizona, 12-17 June 1994.

[3] Brian Goldiez and Russell S. Nelson, "Terrain Database Correlation", presented at ITEC in The Hague, Netherlands, April 1994.

[4] Dr. Richard Economy, Robert Ferguson, and Dr. Eytan Pollak, "Geographic Position Accuracy in a Distributed Simulation Environment", Proceeding of the IMAGE VII Conference, pp. 258-271, Tucson, Arizona, 12-17 June 1994.

[5] Brian Goldiez, Simulator Networking Handbook, Institute for Simulation and Training Doc. IST-TR-93-08, Univ. Central Florida, March, 1993.

[6] James D. Basinger and Stephen D. Ingle, "Database Requirements for Full Mission Simulation", Proceedings of the 1977 Image Conference, pp. 24-33, May, 1977.

[7] Thomas W. Hoog, "Correlated Data Bases for the Present and Future", AIAA Flight Simulation Technologies Conference, pp. 73-78, Arlington, Texas, Sept. 18-20, 1978.

[8] Barbara Townsend, Beth Stovall, and Cheryl Colbert, "Correlation of Sensor Data Bases in the Full-Mission Training Simulator", 7th I/ITSEC Proceedings, pp. 84-90, Orlando, Fl, November 19-21, 1985.

[9] Maureen Hrabar, John Joosten, and Patricia A Widder, "Data Base Conversion/Correlation Issues",, 11th I/ITSEC Proceedings, pp. 343-348, November 13-16, 1989.

[10] Dale H. Fawcett, "Sensor Data Base Correlation", 13th I/ITSEC Proceedings, pp.106-112, Orlando, Fl, December 2-5, 1991.

[11] Karl A. Spuhl and David A. Findley, "Correlation Considerations in the Simulation Environment", Proceeding of the 1994 IMAGE VII Conference, pp. 300-306, Tucson, Arizona, June 12-17, 1994.

[12] Steven R. Schwalm, "Correlation of Environmental Databases for Networked Simulators", Second Conference on Standards for the Interoperability of Defense Simulations, Vol. III, Institute for Simulation and Training Document IST-CF-90-01, University of Central Florida, January, 1990.

[13] Budimir Zvolanek and Douglas E. Dillard, "Database Correlation Testing for Simulation Environments", 14th I/ITSEC Proceedings, pp.867-875, Orlando, Fl, November 2-4, 1992.

[14] Budimir Zvolanek and Douglas E. Dillard, "Quantitative Correlation Testing from DOD Project 2851 Standard Simulator Data Bases", 15th I/ITSEC Proceedings, pp.829-836, Orlando, Fl, November 2-4, 1993.

[15] Richard Dunn-Roberts, Curtis Lisle and Bruce McDonald, "Recommended Approach for Standardizing DIS Terrain Data Bases", Summary Report of the Sixth Workshop on Standards for the Interoperability of Defense Simulations, Vol. II, Institute for Simulation and Training Document IST-CR-92-2, University of Central Florida, March, 1992.

[16] Louis A. Fatale, James R. Ackeret and Jeffrey A. Messmore, "Digital Terrain Elevation Data (DTED) Resolution and Requirements Study: Phase 2 Report", U.S. Army Topographic Engineering Center Technical Report TEC-0035, July, 1993.

[17] John Ellis, "Terrain Modeling Error Measurement", Summary Report of the Sixth Workshop on Standards for the Interoperability of Defense Simulations, Vol. II, Institute for Simulation and Training Document IST-CR-92-2, University of Central Florida, March, 1992.

[18] Mark Kilby, Glenn Martin, Larry Crochet, Chen Jinxiong, and Tinahui Fu, "Correlation Studies for Loral ADST", IST Technical Report 93-17, June 15, 1993.

[19] Michael E. Ginevan, "Testing Land-Use Map Accuracy: Another Look", Photogramm. Engr. Rem. Sens., Vol. 45, No. 10, pp. 1371-1377, Oct., 1979.

MODELING THE CLOUD ENVIRONMENT IN DISTRIBUTED INTERACTIVE SIMULATIONS

Maureen E. Cianciolo
TASC
Reading, Massachusetts

Brian Soderberg
LORAL Advanced Distributed Simulation
Bellevue, Washington

ABSTRACT

This paper describes an ongoing effort to develop and integrate an empirical cloud model within a Distributed Interactive Simulation (DIS) environment in support of high-fidelity training and simulation applications. TASC is developing a cloud model (known as the Cloud Scene Simulation Model) to simulate realistic high-resolution cloud features within domains defined by larger-scale weather conditions. The cloud model generates four-dimensional (three spatial and time), multi-layer cloud fields using a combination of stochastic field generation techniques and convection physics, where internal model parameters are tuned to fit observed cloud data. One data set is generated for each specified output time and contains cloud water density values arranged on a regular volumetric grid. A typical output field contains tens of thousands of data points covering simulation domains of 50 km or more.

Because these data sets are too dense to be transferred across the DIS network or rendered in real-time, we have developed an approach that approximates the complex cloud formations generated by the model as a series of geometric primitives. The cloud data sets are filtered to the level-of-detail appropriate for a particular simulator. The approach uses a planar-wise approximation of a volumetric phenomenon that takes advantage of today's state-of-the-art image generator hardware. The cloud model runs in real-time, allowing for smooth transitions as the weather conditions evolve over the simulation domain.

In this paper, we present an overview of the Cloud Scene Simulation Model (CSSM); its inputs, outputs, and overall methodology. We describe a DIS architecture which enables distributed real-time calculation of large cloud fields, and address usage of and extensions to the standard DIS network protocol. We follow with a description of the volumetric rendering techniques employed in this effort. Finally, we summarize and briefly discuss the application of our methodology to other atmospheric phenomena in future implementations. We conclude our oral presentation with a video tape showing real-time cloud field generation and visualization within a DIS training environment.

ABOUT THE AUTHORS

Maureen Cianciolo is a member of TASC's System Sciences Division. She has over six years of experience in atmospheric physics, software development, and data visualization. As principal investigator of the atmospheric modeling and visualization efforts at TASC, she is currently using her background in physics and computer modeling while leading development of high-resolution models of atmospheric moisture.

Brian Soderberg is a LORAL Division Scientist and the manager of the Computer Image Generator (CIG) development and algorithm research group. Mr. Soderberg has ten years experience in computer graphics and image generator research and development. His research and development focuses on development of new algorithms for image generation to support LORAL's ongoing CIG development and DIS-related programs.

MODELING THE CLOUD ENVIRONMENT IN DISTRIBUTED INTERACTIVE SIMULATIONS

Maureen E. Cianciolo
TASC
Reading, Massachusetts

Brian Soderberg
LORAL Advanced Distributed Simulation
Bellevue, Washington

MOTIVATION

Atmospheric moisture in all of its forms (humidity, precipitation, clouds, etc.) can have significant effects on many aspects of military and commercial systems. However, current simulators are generally limited in their ability to provide a high-fidelity weather environment in order to fully simulate these effects. In this section we briefly outline some of the effects of weather on the operation of sensor systems and local terrain conditions. We discuss the inability of raw weather data alone to satisfy the fidelity requirements of some sensor and terrain models, and point out the challenges of real-time weather data collection.

All of these discussions serve as the motivations behind our ongoing effort to develop a computationally-efficient atmospheric moisture model (consisting of cloud, rain, and humidity models, although we discuss only the cloud model in this paper) that can function with a wide range of input data, generate scenes consistent with these inputs, and provide the necessary level of fidelity for typical sensor and dynamic terrain models.

Weather Effects on Sensors

Atmospheric moisture can impact the operation of military and commercial electro-optical sensors through absorption, scattering, and emission of radiation, and modification of the local atmospheric microphysics (such as aerosol particle size distributions). Examples of these impacts include: variations within moisture fields (e.g., near cloud edges) introducing background clutter that can severely degrade target detection and acquisition; the presence of water vapor attenuating electro-optical signals, thus reducing visibility and transmission; the presence of rain regions which are completely opaque to most common sensor wavelengths.

These atmospheric effects are ubiquitous and can represent a tactically significant tool to a well trained staff on the battlefield. Including these highly variable and highly local effects within the Distributed Interactive Simulation (DIS) training environments provides the opportunity to understand the effects of weather on specific systems and potentially take advantage of them.

Weather Effects on Terrain

The weather affects terrain and terrain-based models in two primary ways: 1) by modifying the radiation received at the ground, thereby introducing temperature and illumination differences across the ground domain, and 2) by modifying the ground characteristics (e.g., trafficability) directly in the presence of rain, snow, ice, or fog. The highly local nature of these effects can be tactically significant and therefore should be accounted for in realistic training environments. Along with the spatial distribution of the weather parameters, their temporal distribution is important. Ground force and ground vehicle simulators need to know the time-history of any precipitation to update soil characteristics and respond accordingly.

Weather Data Availability

Much of the raw weather data that are readily available (including temperature, moisture, pressure, wind, cloud layer information, rain rates, etc.) come from national or global networks of numerical weather prediction models, surface observations, and upper air balloon measurements. The resolution of these data sources is very coarse (on the order of tens of kilometers or more), which, although useful to characterize the general state of the atmosphere on a regional scale, does not capture the higher-resolution features within local regions and therefore does not satisfy the fidelity requirements of many simulators. (A limited number of higher-resolution meteorological data sets exist in the vicinity of some airports or in support of special military training exercises, however, they are not generally available.)

In addition, we must consider the operational availability of these data to support war-time mission planning and rehearsal. During periods of conflict only limited weather data may be available due to data black-outs, limited communications, etc. Training/planning systems being used and developed for these conditions cannot depend on using special databases or full complements of meteorological data.

CLOUD MODEL OVERVIEW

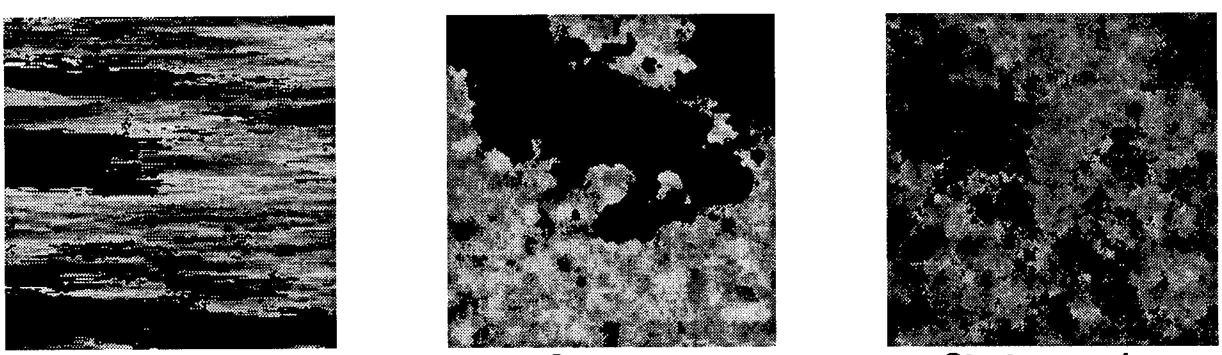
Methodology

The Cloud Scene Simulation Model (Ref. 1) is an empirical model that generates high-resolution, four-dimensional (three spatial and time), multi-layer (low, middle, high, and convective layers) cloud fields consistent with larger-scale input weather conditions. That is, it simulates realistic structure (typical resolutions of 1-100 meters) within a domain defined by general meteorological characteristics. One output field is generated for each specified output time and contains cloud water density values arranged on a regular volumetric grid. The model can simulate a variety of cloud types including: cirriform (high, thin, cloud streaks), stratiform (low, homogeneous cloud layers), and cumuliform (puffy, vertically-developed convective clouds). Examples of three different cloud types are shown in Figure 1. Terrain-induced cloud types, including fog and roll clouds, will be added in the near future.

The model uses a fractal algorithm (the Rescale and Add algorithm, Ref. 2) to specify the horizontal distribution of cloud elements across the user-specified model domain, where parameters within the fractal algorithm are tuned to fit observed cloud data (e.g. the variability of liquid water density within cloud elements of differing types is controlled by a "length" parameter within the algorithm which is determined by an analysis of aircraft-based cloud measurements). The vertical growth of the clouds is modeled using convection physics and knowledge of atmospheric structure. Comparisons with real data have shown that the model captures the characteristically complex internal and external structure of cloud fields observed in nature.

The fractal algorithm within the model controls the stochastic distribution of the clouds. It is initialized with a number seed. By using the same seed and initialization data, one can reproduce identical scenes and ensure that individual simulators use identical cloud fields. By varying the number seed, one can simulate large numbers of unique fields, all consistent with the input weather state, which could then be used for Monte Carlo studies.

The model simulates three-dimensional cloud fields evolving in time. For simulations evolving in time, the model uses the bounding weather conditions at the beginning of the time period and the end when determining intermediate cloud fields. For example, a high-resolution simulation may require cloud fields every 1 minute although weather data (from a forecast model or archived data base) may only be available every one hour. The model correctly "interpolates" for all time periods in between taking into account temporal evolution (growth, dissipation, and diffusion are simulated using a fractal algorithm) as well as advection (cloud motion due to local winds).



Cirrus **Stratus** **Stratocumulus**

Figure 1 Two-dimensional slices (nadir view) of sample cloud types generated with the CSSM

The model is written in ANSI C. It currently runs in two modes: 1) stand-alone mode in which it generates sequences of data files that are then rendered independently, and 2) DIS-integrated mode in which it is initialized with an environmental PDU (Protocol Data Unit), and then simulates fields that are rendered within a CIG. The integration of the cloud model within a DIS environment is the primary focus of the next section. But first we provide more detail concerning the input and output of the cloud model.

Model Input

The model simulates cloud fields at very high resolutions (on the order of 1-100 meters) consistent with coarser-resolution user-supplied meteorological input data. Along with providing the meteorological inputs, the user also defines the simulation domain by specifying an (x,y,z,t) origin, spatial extent and resolution, and a temporal period and resolution. (Within the DIS-integrated mode, these parameters can vary during the simulation, e.g., the origin can vary as the region of interest changes.)

The model can use single-valued inputs over the region of interest. For example, a user simply specifies overall cloud layer information (e.g., 50% stratus layer at an altitude of 1000 meters and 80% cirrus layer at 3500 meters) and a single wind, temperature, and relative humidity vertical profile for the simulation domain. Or the model can be initialized with gridded inputs where cloud layers and meteorological variables can vary across the simulation domain. In either case, the model generates a realistic higher-resolution scene that satisfies these input criteria.

In our DIS-compatible implementation, we provide all model inputs through an environmental PDU. The structure of that PDU is based on the environmental state PDU proposed at the most recent DIS Workshop Simulated Environment Working Group (Ref. 3). We added to it the parameters required to initiate the CSSM software. (The PDU extent field designates the number of extra parameters required for running the cloud generator. See Table 1.) A list of the parameters required specifically by the CSSM follows:

Meteorological data (gridded or single-valued)
- wind as a function of altitude
- temperature as a function of altitude
- dewpoint temperature as a function of altitude
- pressure as a function of altitude

Cloud layer data (gridded or single-valued)
- cloud amount (0-100%) by layer
- cloud type by layer
- mean cloud base height by layer
- maximum cloud top height by layer

Domain parameters
- domain origin (x, y, z, and time)
- domain extent (e.g., 100 x 100 kilometers)
- domain resolution (e.g., 10 meters)
- temporal extent (e.g., 60 minutes)
- temporal resolution (e.g., 3 minutes)

Miscellaneous parameters
- root filename for output files
- number seed (to initialize the fractal algorithm).

The input parameters concerning the location of the domain origin (in space and time) have been included to satisfy simulations that move over time or cover a large spatial extent (beyond any single simulator's field of view). In these cases it is not necessary to model the atmosphere over the entire simulation domain, but instead just over the region of interest to each simulator. The domain origin parameters ensure that all the simulation participants see a consistent and continuous cloud environment which is paramount in DIS applications.

Model Output

The Cloud Scene Simulation Model generates a single output file for each user-specified output period. The files contain liquid/ice water content (LWC) values (grams/m^3) at every point in the user-defined model domain. That is, each output file is a three-dimensional snapshot of the scene at a given time. Visualization of these fields within a DIS environment is the subject of a later section. How these synthetic output fields compare with actual cloud fields is the subject of the next section.

Comparison to Real Cloud Data

A number of studies have shown that clouds show characteristic fractal behavior over a broad range of spatial scales (Refs. 4, 5). We use the Rescale and Add (RSA) fractal algorithm (Ref. 2) within the model to simulate the spatial distribution of cloud cover and liquid water content. A number of parameters within the RSA algorithm control the statistical characteristics of the resulting fields. In our previous modeling effort, we used a limited set of cloud observations to tune these fractal parameters by comparing LWC time series sampled using conventional hot-wire probes mounted on aircraft to series sampled from the model output fields. Figure 2 shows two sample cases. We are currently collecting a large number of additional observations spanning a wide range of climatological conditions for further analysis. We will use a portion of these observations for parameter estimation and set aside the remainder for model validation.

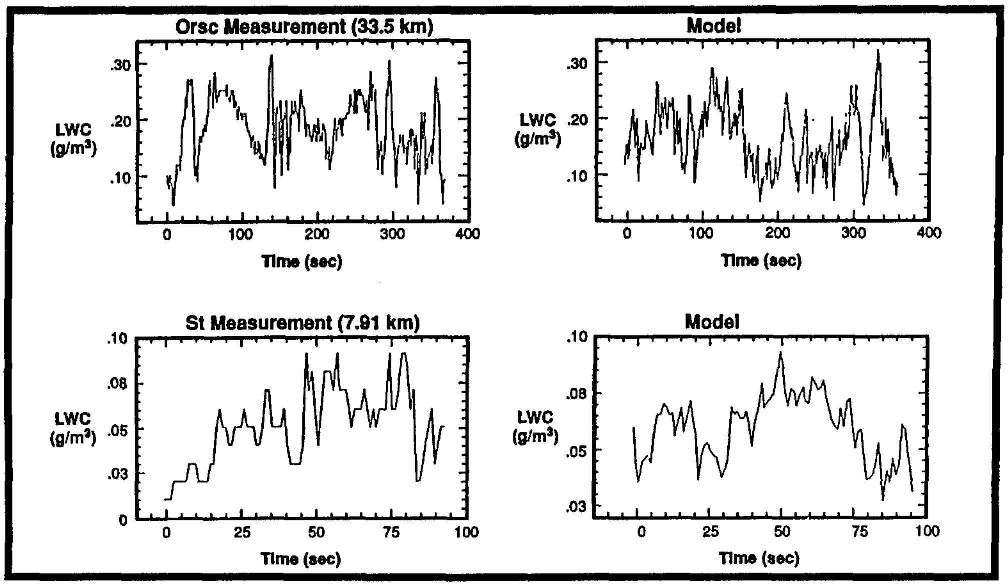


Figure 2 Comparison of measured and model-produced LWC spatial series for two different cloud types: OrSc (orographic stratocumulus) and St (stratus)

CLOUD SIMULATION WITHIN A DIS ENVIRONMENT

This section briefly presents the traditional DIS simulation node architecture, followed by a detailed description of the working architecture developed under this effort which allows integration of the cloud scene simulation and visualization models.

Standard DIS Architecture

Figure 3 presents a logical view of processing at a simulator node on the DIS network. Information on the DIS network is exchanged by way of Protocol Data Units (PDUs). PDUs describe the position and state of vehicles and other dynamic entities on the network. The *simulation host* is responsible for dynamics simulation and processing messages sent over the network. The host sends the computer image generator ownship vehicle data such as position, rounds fired, etc. as well as positional updates of other dynamic entities on the network to be displayed. The front end of the CIG is the *graphics processor* which is responsible for processing messages from the host, managing the viewport configuration, paging data from disk, and sending commands and data to the graphics hardware pipeline. State-of-the-art graphics workstations, such as the SGI Onyx, allow the simulation host and graphics processor to perform on common processors, blurring the distinction between these two logical functions.

We use this same basic architecture in our implementation with a number of extensions to support real-time cloud visualization. We describe those extensions in the next section.

DIS Architecture with Clouds

The CSSM requires substantial time (up to 90 seconds for a typical scenario) to calculate a single cloud data field. Typical DIS simulations run at 15 to 30 frames per second. To accommodate the CSSM processing time, the system architecture was designed to allow for concurrent synchronous (with frame rate) and asynchronous dynamic environmental model processes to run within the DIS simulation. The CSSM runs in the background storing its output

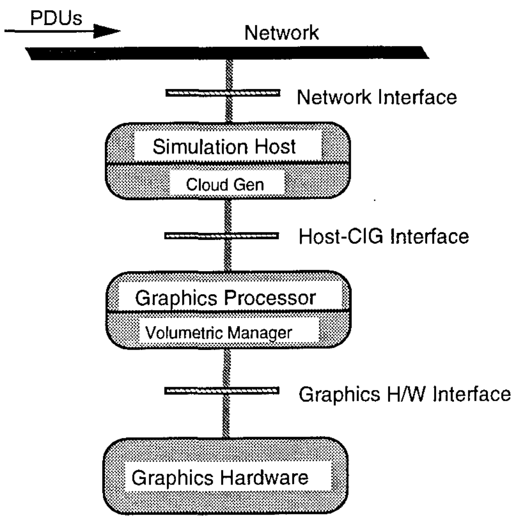


Figure 3 Logical view of DIS node processing

data in shared memory for subsequent processing. With each frame, the Volumetric Primitive Manager performs conversion, integration, and sorting of the volumetric cloud data with other synchronous model primitives prior to developing graphic element calls for visualization. This must be done to accommodate the graphics requirement to sort all transparent objects and render them back to front. (This is specific to the SGI Onyx Graphics Workstation.) This basic architecture, with the functionality to allow concurrent synchronous and asynchronous processes, should allow future incorporation of new models requiring volumetric simulation and visualization.

Figure 4 shows the entire data flow through our DIS-compatible cloud visualization system (the system also simulates smoke plumes, but we focus on cloud simulation in this paper). Data flow begins with PDUs, continues onto cloud modeling and the volumetric primitive handling which generates the hardware specific graphics calls. Each of the major components of the system is described in the following subsections.

Cloud Generator (CG). The CG executes under the control of the simulation host and is built around the Cloud Scene Simulation Model. The CG uses input cloud parameters specified in an environmental PDU, sounding data, and other parameters which depend on the type of simulation host (e.g., ground or air vehicle). It generates a temporal sequence of three-dimensional cloud fields as output. Interpolation between any two sequential cloud fields in the CIG results in smooth translation from one time to the next, and therefore results in smooth cloud motion and evolution.

Note: An alternate approach (not implemented for this effort) is for the CG to execute on a server and pass the cloud model data to individual simulators over the network. This approach reduces the local computation at nodes at the expense of increased network traffic.

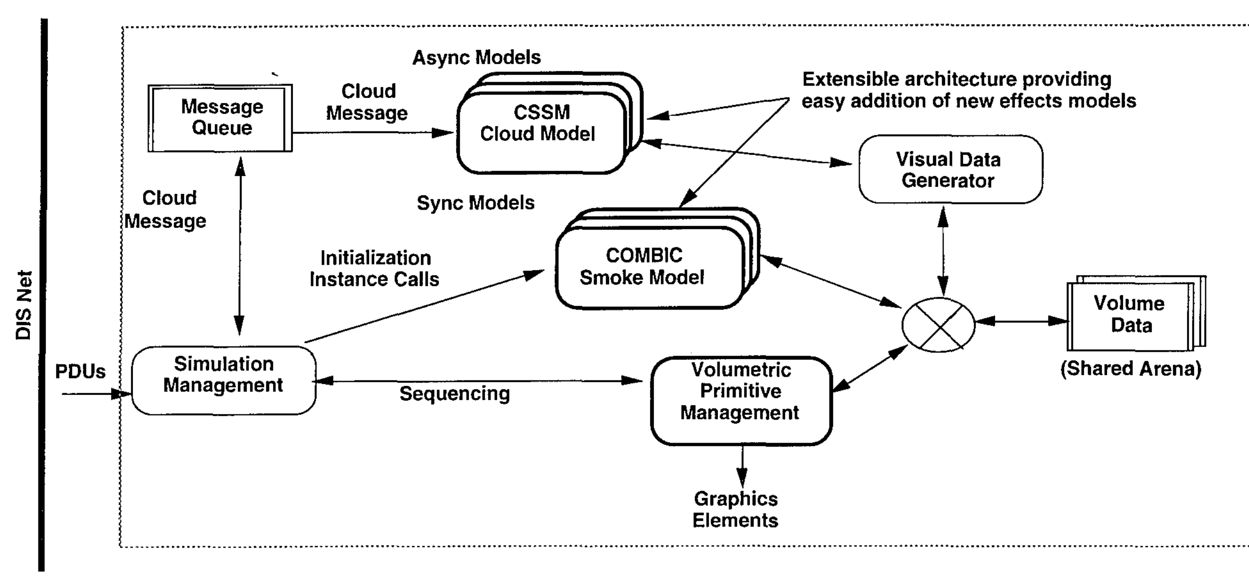


Figure 4 System data flow with integrated cloud simulation and visualization

Visual Data Generator (VDG). This module uses two sequential cloud fields to create the required data structures for the visualization of the clouds. It performs the following functions:

1) Calculate interpolation data for each primitive position, opacity, and dimension to provide the volumetric manager the capability to smoothly interpolate at up to 30 frames per second for smooth visualization.
2) Add random perturbations in position, dimension, and density of the graphics primitives used to represent the gridded cloud field. This produces a more realistic rendering of an otherwise regular Cartesian volume grid. (The number seed used to initiate the random perturbations is stored in the environmental PDU to ensure that all DIS nodes calculate the perturbations in an identical fashion so primitives are fully correlated across DIS applications.)
3) Set the primitive ambient and diffuse lighting components based on the relationship of each primitive with other primitives and the global illumination vector. This allows for self shadowing of the cloud primitives.

Volumetric Manager (VM). The VM executes on the CIG under the control of the graphics processor. The VM manages the loading of volume data and interpolation of the current field between two time periods. The VM then sorts the data primitives back to front to allow for graphics processing.

Network Data Interface. An entity on the network, known as the environmental manager, is responsible for controlling the overall environment during a DIS exercise. Periodically, the environmental manager issues an environmental PDU specifying cloud model parameters. These parameters include cloud type, fractional coverage, base and top layer heights, along with the simulation domain extent and resolution, etc. (See Model Input section presented previously.)

Graphics Hardware Data Interface. The cloud data primitives are displayed as two-dimensional textured or vertex-colored polygons with variable size, opacity, and intensity. The CIG's graphics library calls define the interface between the graphics processor and the graphics hardware. These calls are platform dependent.

GRAPHICS RENDERING PROCESS

One of the critical components to this effort was to find an efficient approach to rendering volumetric phenomena on off-the-shelf graphics workstations or image generators. Current DIS software packages are good at representing surfaces, but not very good at representing fuzzy volumetric phenomena. Our challenge was to come up with a *polygonal method* to approximate volumetric elements required for the cloud simulation and visualization. That method is described in the following subsections, where we begin with a discussion of the method for approximating the overall cloud structure and follow with an outline of the lower level graphics primitive implementation.

Cloud Structure

The requirements for the high level cloud geometry structure are: 1) it must be derived from the CSSM software 2) it must allow interoperable visibility between any two vantage points to ensure a fair fight among DIS users, and 3) it must accommodate varying levels of data resolution to support varying levels of fidelity graphics hardware and database configurations.

There were two basic choices for the graphics geometry representation. The first involves use of a complex polygonal hull to describe the cloud shape or structure. This method was deemed unresponsive to the requirement for interoperability. Although providing realistic surface representation while outside the cloud structure, a polygonal hull exhibits improper intervisibility when two vehicles are viewing each other from within the cloud. The second method (selected for implementation) is described in the following section.

Geometric Primitive Aggregation. This approach groups many small graphics primitives together to provide the total cloud structure. For example, a large number of small spheres, each with their own individual attribute settings, could be used to form a complex cloud structure.

Advantages-- The primary advantage of such a method is that the overall look can be very accurate with many small primitives. Each primitive, with its own attribute settings, can accurately model shape, lighting, and density variations throughout the cloud domain. In addition, it can be accomplished on today's graphics workstations such as the SGI. The resolution of the primitives can be adjusted for varying levels of fidelity and processing speeds. The viewing interoperability requirement is satisfied with this technique.

Disadvantages-- The primary disadvantage of this method is that at high primitive density, a large number of polygons is required. When looking through many layers of primitives, the pixel processing load can become excessive. Hence, some form of pixel processing control is required. We plan to implement a simple level-of-detail control to handle this problem in the future.

A high-level schematic of the rendering process is presented in Figure 5. That figure shows the structure of the cloud field throughout the visualization process. As shown in the upper left, the CSSM generates a three-dimensional regular grid of LWC data. These data are converted to spherical visual primitives for graphical representation with associated radius, location, opacity, and lighting characteristics. Random perturbations to the location, radius, and opacity of the primitives are then imposed for additional realism. Self shadowing is later added.

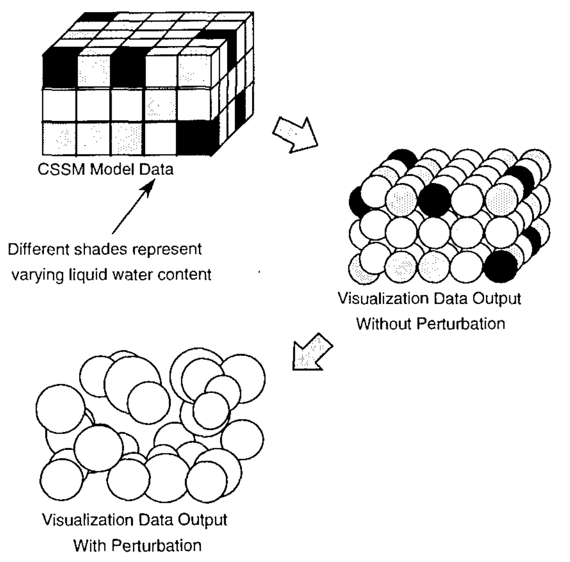


Figure 5 Cloud model data structures within the visualization process

Low-level Graphics Primitives
Under this effort, we developed a method of aggregating many simple low-level primitives to perform a planar-wise approximation of volumetric phenomena. The previous section discusses that approach. This section describes the attributes of the low-level primitives that we selected. The primary requirements that drove our selection were: 1) real-time processing speeds, 2) realistic visual representation, and 3) minimal intervisibility problems.

Volumetric Primitive Attributes. We considered several different primitive types in our implementation. Some of these are shown in Figure 6. Each of the base geometric primitives includes attributes to allow realistic visual processing. They include color or texture pattern, opacity, shading attributes (flat or smooth), and material type to support sensor visualization. We selected angle-oriented disks and polygons for our implementation as they both consist of the fewest number of polygons and produce a realistic visual effect.

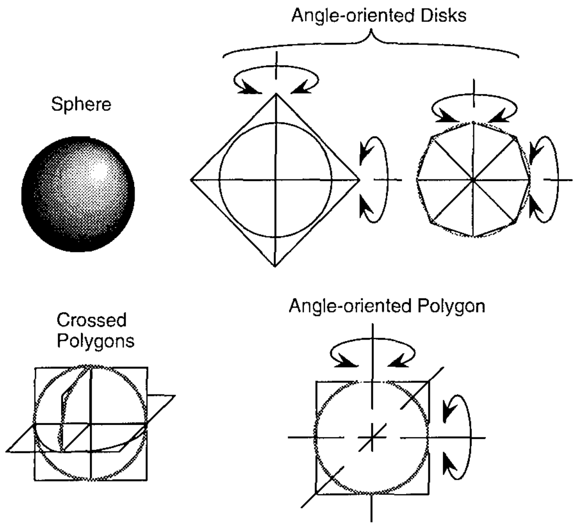


Figure 6 Several approaches for the base volumetric primitive geometry for cloud visualization

Self Shadowing. The shading parameters for each cloud primitive are calculated as a function of the number of cloud primitives between it and a light source and the density of the cloud field along that path. This results in variable shading across the cloud field which produces a more realistic visualization.

RESULTS

The two photo plates (Figs. 7 and 8) show output from the cloud visualization system. Both scenes show cumulus cloud layers from different viewpoints: an airplane view (Fig. 7) and a low-flying helicopter view (Fig. 8). Both scenes were generated in real time on a Silicon Graphics Onyx deskside workstation.

SUMMARY

In this paper we describe an ongoing effort to develop a high-fidelity cloud visualization system to provide physically-based dynamic environmental effects in DIS applications.

Under this effort with Phillips Laboratory and TEC, the CSSM software has been successfully integrated into a networked simulation software package. In addition, a new volumetric primitive management method has been developed to visualize dynamic cloud entities in real time. (The overall architecture has also been extended to simulate other physics-based models such as smoke plumes. See Ref. 6.)

The resulting DIS visualization architecture will be used in future dynamic environment modeling experiments to support a variety of simulation efforts such as upgrades to the Modular Semi-Automated Forces (ModSAF) simulation. These ModSAF upgrades will include simulating rain, dust clouds, smoke plumes, and other environmental weather conditions along with clouds to determine their effects on automated forces behavior. We believe that the basic modular architecture described in this paper will be extendible to those experiments.

ACKNOWLEDGMENTS

Special thanks to Jeff Turner, Chris Moscoso and George Lukes, of the US Army TEC, and Don Grantham of the US Air Force Phillips Laboratory who supported, advised, and sponsored this DIS integration and development effort.

REFERENCES

[1] Cianciolo, M.E., and R.G. Rasmussen, "Cloud Scene Simulation Modeling - The Enhanced Model," Technical Report, PL-TR-92-2106, April 1992.

[2] Saupe, D., "Point Evaluation of Multi-Variable Random Fractals," Visualisierung in Mathematik and Naturwissenschaft, H. Jurgens and D. Saupe, Eds., Springer-Verlag, Heidelberg, 1989.

[3] Beverina, A.F., and J.R. White, "An Alternative Environment PDU for Initiation, Control, and Hand-off of Embedded Processes," 10th DIS Workshop, March, 1994.

[4] Lovejoy, S., "Area-Perimeter Relation for Rain and Cloud Areas," Science, 216, 185-187, 1982.

[5] Cahalan, R.F., and J.H. Joseph, "Fractal Statistics of Cloud Fields," Monthly Weather Review, 117, 261-272, 1989.

[6] Soderberg, B.T., and J. Gardner, "Dynamic Environment Modeling in Distributed Interactive Simulation - Final Report," June, 1994.

Figure 7 Real-time cloud scene generated with the CSSM and volumetric processing software (view from above a cumulus cloud layer looking down toward an airport runway)

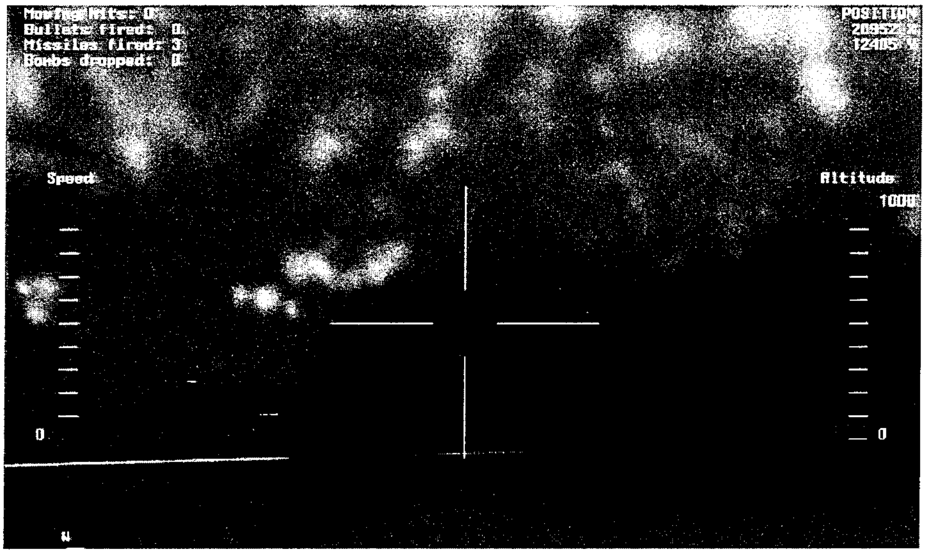


Figure 8 Cumulus cloud layer with cloud shadow processing in a dusk environment. In addition, physics-based smoke plume modeling is added using the same volumetric processing technique

SIZE	FIELD TYPE	DATA	EXPLANATION
96	PDU Header	Protocol Version - 8-bit enumeration Exercise ID - 8-bit unsigned integer PDU Type - 8-bit enumeration Padding - 8-bits unused Time Stamp - 32 bit unsigned integer Length - 16-bit unsigned integer Padding - 16 bits unused	PDU Header is same format as Version 2.0.3 Entity State PDU (ESPDU)
64	Environmental Entity ID	Site - 16-bit unsigned integer Application - 16-bit unsigned integer entity - 16-bit unsigned integer Padding - 16 bits unused	ID is same format as ESPDU
64	Environmental Entity Type	Domain - 8-bit enumeration Kind - 8-bit enumeration Country - 8-bit enumeration Subcategory - 8-bit enumeration Specific - 8-bit enumeration Extra - 8-bit enumeration Padding - 16 bits unused	This type record is unique to the Environment PDU. Specific enumeration for the record is being worked in the Atmospheric Subgroup of the Synthetic Environmental Working Group
192	Location in World	X-Component - 64-bit floating point Y-Component - 64-bit floating point Z-Component - 64-bit floating point	Same as ESPDU
96	Linear Velocity	X-Component - 32-bit floating point Y-Component - 32-bit floating point Z-Component - 32-bit floating point	Same as ESPDU
96	Entity Orientation	Psi - 32-bit floating point Theta - 32-bit floating point Phi - 32-bit floating point	Same as ESPDU
32	Appearance	32-bit record of enumerations	Same as ESPDU
320	Dead Reckoning Parameters	Dead Reckoning Algorithm - 8-bit enumeration Other Parameters - 120 bits unused Entity Linear Accel - 3*32-bit floating point Entity Angular Velocity - 3*32-bit integers	Use Standard dead reckoning algorithms plus new TBD algorithms for embedded process
32	PDU extent	Number of Parameters - 16-bit unsigned integer Size of each parameter - 16-bit unsigned integer	The following allows the user to define parameters for specific entities
96	Weather Information	Number of Sounding Levels - 16-bit unsigned int ID of Sounding Data - 16-bit unsigned integer Random Seed - 32-bit unsigned integer	Sounding data specifies pressure, temperature, dewpoint temperature, and wind as a function of altitude across domain
128	Cloud Temporal Information	Number of Output Periods - 32-big floating point Temporal Resolution - 32-bit floating point Starting Time - 64-bit unsigned integer	Starting time gives the seconds since midnight of Jan. 1, 1970
32	Cloud Layer Information	Number of Layers - 16-bit unsigned integer Padding - 16 bits unused	This number includes cumulus and non-cumulus cloud layers
nx128	Cloud Layer Parameters	Fractional Coverage - 32-bit floating point Base Height - 32-bit floating point Top Height - 32-bit floating point Cloud Type - 8-bit enumeration Padding - 24 bits unused	Cloud layer parameters describe the thickness, cloud type and fractional amount of each layer. At most 4 layers are allowed in the current cloud model

Table 1 Cloud Entity State PDU Definition

MODELING SIMULATION OBJECTS WITH RASP, NIAM AND HCPN

Bo Hagerf
CelsiusTech
Järfälla, Sweden

ABSTRACT

To achieve efficient interoperability – horizontal as well as vertical – between the nodes in a DIS network and thus ensure that the simulations will be truly interactive, it is essential that the various simulation objects in different simulation models are structurally correlated or equal. This is especially important for the development of computer generated forces for computer-based training for different command levels – operational as well as tactical – where the artificial forces, equipped with sensor models, weapon models and internal decision models will give the trainees a realistic though virtual combat and war gaming environment.

The methodology presented in this paper focuses on the means of developing an effective and unbroken traceability chain from computer simulation models that describe the concepts, information objects and constraints and evolves to fully-fledged systems models, back to the real world models that describe and represent the human endeavours, goals and means in warfighting.

The important side effects of this methodology are high quality and practical database designs for tactical battlefield C3I systems and computer-based training that have advanced demands on information consistency and flexibility.

The methods used are RASP (Requirements Analysis and Specification, developed at Chalmers Institute of Technology, Sweden), NIAM (Nijssen Information Analysis Methodology or Natural language Analysis Method, by professor Nijssen, Holland) and HCPN (Hierarchical Colored Petri Nets, developed by professor Petri, Germany and professor Jensen, Denmark et alii).

ABOUT THE AUTHOR

Bo Hagerf is a Senior Systems Analyst at CelsiusTech in Sweden. He has been responsible for systems modeling activities in several projects e.g. the Swedish Air Defense System STRIC, the STRIC's advanced trainer and simulator STRICS, the Swedish Army Tactical Trainer TRACCS, Australian Army Tactical Command Support System AUSTACCS. He holds a degree (similar to an MS) in Mathematics, Logic and Computer Science (University of Stockholm and Royal Institute of Technology).

MODELING SIMULATION OBJECTS WITH RASP, NIAM AND HCPN

Bo Hagerf
CelsiusTech
Järfälla, Sweden

INTRODUCTION

Considerable progress has been made in the design and implementation of computer simulations over the last two decades. Today, computer simulations are important in many application areas, in defense, education, science and industry - non-mention entertainment! Some of the most well-known examples are simulators that try to realistically control and render a computer simulated real world, such as a world of soldiers, air planes, tanks, helicopters, and ships. The purpose and ultimate intention is that e.g. by using real life control panels connected to a computer generated or simulated objects behave as close to their intended real or physical counterparts as possible.

Typically, problems of simulating laws of nature are predominant in such simulation environments. For example, the effect of driving a simulated tank close to a swamp should include considerations of wetness of the ground. Modern simulators also provide multi-user distributed interaction, which raises problems of how each user's interactions with the simulated world are to be synchronized with the actions of the other users.

A different type of simulation is the simulation of dynamic systems of interacting objects. Here the main focus is the study of emergent behavior, i.e. behavior which is inherent in the local laws that control the individual objects. Such systems are characterized by their non-linear behavior, which sometimes makes them unpredictable, but also by the fact that the systems are adaptive to external events. Dynamic systems are ubiquitous, but can typically be found in stock and currency markets, in battle situations, in highly competitive businesses and in different civilian traffic systems (air, naval and land).

As a concrete and simple example of an adaptive dynamic system, consider movement of armored vehicles.

The reason for simulating dynamic systems is to study their behavior and to reveal what behavioral effects different objects reasonings have. Stability, chaotic state, and resource consumption of such systems are important issues which can be experimentally analysed by simulation. The results of the simulations can be used for training, research and decision support.

In education, simulators will be used much more than today for illustrating physical phenomena. Abstract and complex processes can be made concrete, improving the learning curve significantly.

In the natural sciences, some claim that computer simulations will become as important as traditional theoretical and experimental methods. If the performance of the next generations of computer hardware responds to the expectations of today it will become possible to find solutions to a great number of previously intractable simulation problems of physics and engineering. Thus, new branches of research will open up.

In the future simulation will also be used more extensively by companies for leadership training, including market analysis and decision making. There are many experts interested in exploring simulation techniques to a greater extent in the future. Among others economists, who experiment with different simulated models of financial transactions, and social scientists, who study the behavior of simulated societies.

The military has pioneered the use of simulations, and have been utilizing simulators in the training of personnel during several decades. Currently, the use of simulation is focused on tactical and operational cooperative work. Previously, simulators have been based on rendering, as realistically as possible individual experiences. In addition, the cooperation between individuals and groups will be emphasized in the future. Hence, collaborative efforts will be practised cheaper and with better control in the simulation laboratory, thus improving modern warfare tactics and execution.

The full simulation of complex, real-life objects which act autonomously and communicate asynchronously in a dynamic environment, presupposes that systems where each object behaves:

- interactively;
- reactively;
- adaptively;
- didactically;
- autonomously;

will be created.

MODELING AND SIMULATION

Humans solve problems and understand the world through models i.e. human-constructive formal structures that can be communicated and understood through a process of interpretation.

The models we build must be tested, which means that we have to create an environment and a scenario where we simulate the real world.

Sufficiently Understood?

One problem that most people have is to be able to identify the purpose of an model and at the same time be aware of the inherent limitations every model brings.

Some models are used under circumstances or assumptions for which they were never intended. The consequences of forgetting these model constraints can be enormous.

The operational and tactical situation picture is of vital importance to the military decision-maker. When war performance was of small or limited proportions, regarding geographic extension as well as number of combatant units, commanders could take in the situation directly and convey their decisions to the affected units via orderlies.

The modern warfare is of a different nature. In many respects, it is total. Operations separated in time and space are dependent on each other in an intricate manner. They often make use of the same resources. Speed, destruction and elimination effectiveness etc. have all been multiplied. The commander has no possibility to directly observe the area of operations. Ranges and speeds of the new weapon types are beyond the observation capability of the individual commanders. Instead, focus is on synthetic situation pictures. They are created by means of information, messages and reports from a number of sources. The effectiveness and the ranges of sensors and weapon systems have increased. Also the quantity and quality of the information has grown in pace with the development of different types of sensors – a virtual information explosion.

Therefore, it is very important to try to aggregate and filter the information so that the decision-maker will not be paralysed by all the information. However, the principles for doing this are not clear. At the same time, it is important that the command system is open and transparent; if the decision-maker wants more detailed descriptions of the situation, it must be easy to satisfy this increased information requirement.

Challenges For DIS

DIS in itself is a model, a model for distributed interactive simulation. To make use of DIS under assumptions that are not compatible with the DIS concept can be devastating for the whole concept. Moreover, which is perhaps even more important, the simulation objects must have a comprehensible structure for the military and the systems analyst. Therefore it is of vital importance to have an encompassing model for all affected categories involved. Without an all-encompassing business model, a concise information model and a system model it will be hard to keep the intended system structure complete and consistent.

For each on-line simulation in a DIS environment there are some questions that have to be answered e.g.:

- What is the resolution of information for each interactive operator? (there is no reason to assume that all operators have the same needs of resolution of information)
- What is the granularity of the information operators exchange?
- How often is each kind of information updated?

Support for Adequate Systems Architectures

Distributed and concurrent software as well as operating systems support are very important features of the systems architecture that has to be used.

The most suitable architecture for DIS seems to be an Object Management Architecture (OMA). To achieve this goal, it is most important to identify the objects in a manner that gives the stability and traceability of a formal syntax and language.

A generic open command support system must provide both for reusable common capabilities and high degrees of functionality through some seamless integration of commercial off-the-shelf components. In all these system components the object model of all included concepts must be well stated and clear, even if the implementation of an object varies between the different system components.

MODELS AT DIFFERENT LEVELS

Three Steps – Three Methods

In order to design and define a system (simulation or command support system), it is necessary to decide for whom, to what purpose, in which organizational structure, to what use etc. the system is intended. We'll have to determine all objects and part-of objects, whether they consist of other objects and describe the behavior and the constraints and rules they are modeled to follow.

Therefore it is essential to have a consistent and complete model of the world of discourse expressed in its own terminology without any technicalities.

First, a business, enterprise or job model is important.

Secondly, an all embracing information model that identifies on a formal ground all information objects and their mutual constraints.

Thirdly and finally, a formal system model that embraces all dynamic behavior, concurrency, resource competitiveness, etc.

These three steps are initially in sequence, but after the first iteration, the steps will be in parallel.

Which methods can support this process and what selection method is most suitable? A formal mathematical foundation guarantees the soundness of these consecutive steps, and is therefore preferred, even if formal methods can be hard to learn and somewhat cumbersome to practice.

Many systems engineering tools today lack firm foundations, many are but fancy drawing tools. It is possible to create a number of elaborate drawings, but after some time it is very cumbersome to handle this material and the benefit is, to say the least, near null.

The three methods briefly presented below are however of a different nature. Their foundations are formal and rigid. (The last two methods are even mathematically based.)

RASP

RASP is a methodology, that is a collection of methods and computer support, for business analysis. All aspects of a business is handled with this method, both formal and informal perspectives. RASP stands for Requirements Analysis and Specification and was developed at Chalmers Institute of Technology, Gothenburg, Sweden.

Modern organizations, whether military, other public authorities or business enterprises, live in a world of constant change. Politicians, the public at large, customers, suppliers, financiers etc. place demands on the services and products, the efficiency of Defense, environmental effects, community service, etc. It is difficult to satisfy these demands, especially when they are often in conflict with each other. Most difficult of all is to put all the parts together to make a well-balanced, efficient whole. Rapid technological development, tough competition in the marketplace, changing economic cycles, new laws and government restrictions require adaption, restructuring, growth or cutbacks. In order to face and master these challenges continuous development and adaption is required. The single most important aid to manage this task is a well-integrated methodology for organizational change and development.

Business analysis according to RASP is not to be reduced to a biased technical or real-procedural description of the buisness domain. Adequate attention to informal aspects (e.g. traditions, customs, heuristics) are emphasized.

NIAM

Nijssen Information Analysis Methodology or Natural language Information Analysis Method is a method to identify and analyse the information objects of a delimited systems domain. The method gives a genuine semantic model based on the natural language with a basic distinction between things and names-of-things, which is common when we want to be precise. NIAM is a discipline of information analysis necessary to to uncover the syntactical-semantic structure of objects.

There are several methods for information analysis that uses conceptual schemes. However NIAM is the most semantically powerful method in its precision, well-stated foundations and good formalism. At the same time, NIAM, due to the natural language basics, lets the end-users and application experts, e.g. military, formulate their respective experiences and views in their own words and concepts. This is one of the reasons why a business model done in RASP, can (at least partially) be used as initial in-put into NIAM.

The basic information construct in NIAM is the object-type and the associations between them, the so called fact-types, and the way to refer to an object-type by giving it one or more names or label-types. In all associations between object-types the role(s) identifying it must be specified. The same is true for all reference-types. An absolute requirement is that every object-type must have at least one unique reference - identification or uniqueness constraint. Several other constraints are supplied in NIAM such as equality, exclusive, subset, total, joint total, subtype and range.

From a NIAM model it is relatively easy to accomplish a normalized RDBMS schema. With a suitable NIAM modeling tool (e.g. RIDL, see below) this schema can be automatically generated.

HCPN

Hierarchical Colored Petri Nets (HCPN) is a method for systems analysis that has been developed since the early sixties. The HCPN language is very suitable for providing modeling and simulation of complex systems, e.g. command and control (C3I) systems, computer-based training and simulation of intelligent objects.

The Petri net formalism concurrency, non-determinism, dead-lock, etc. Furthermore a HCPN model is ultimately equivalent to executable computer program. The inherent control structure of a HCPN model manages the synchronization aspects and enables the systems analyst to concentrate on more detailed issues and local modeling problems.

In HCPN one can model all different dynamical aspects of a complex system. Futhermore it is possible to express the details of the systems model at an arbitrary level of detail, thus render a true top-down design possible.

Furthermore the HCPN formalism and the supporting tools give the analyst an unique opportunity to make performance analysis before building the actual system or implementing a simulation object for distributed simulation.

The mathematical theory (graph theory) of HCPN is valuable guarantee of the correctness of the method comparable to that of NIAM (symbolic algebra).

Representing Knowledge

The three methods support each other and take into focus different aspects of the universe of discourse. It is of high importance to all military models that they in some way include these different but complementary model perspectives. To represent e.g. a battalion staff and all the information that is used in a staff, the information objects, the functional dynamics and all the constraints that are attached to that domain must be adequately modeled.

The three methods together enable the systems analyst and systems designer to construct a formal rigid model of the different objects of the system intended. Furthermore, these formalisms from business model, information model to dynamic and executable system model guarantee a traceable chain of models. Thus ensuring manageble systems support to modify the simulation objects, i.e. if the real-life objects behavior alters, the computer system objects changes correspondingly.

Tool Support

The three abovementioned methods have computer support. RASP has an efficient tool that is developed in Sweden for the Macintosh, MacRASP. NIAM is supported by several tools on the market, perhaps the most eminent tool has been developed in Belgium, RIDL and the U.S. company Meta Software has produced a very good tool for HCPN, Design/CPN.

Tool Integration

There is however no fully-fledged tool integration yet. The different formalisms are not exchangeable. The reason is that they model the world of discourse from three different but supplementary or complementary perspectives. Together they can give covering design of the simulation objects intended.

However, the results from the RASP model can be used as an input to NIAM and conversely. Similarly a NIAM model is indeed very useful during the construction of an HCPN model.

METHODOLOGICAL CHAIN

Description

From a RASP model of e.g. an armored battalion, you filter out all the notions and concepts in that model and start to analyse all of them to get a clear picture. The resulting NIAM model will encapsulate everything. Now it is time to take the dynamics of the system into consideration. You build an HCPN model. The main advantage with that kind of model is that it enables you to bring about all constraints and dynamics of the concepts you intend to construct.

Experiences in Projects

At CelsiusTech we have used RASP and NIAM for a long time in several projects. HCPN, however, has been introduced lately.

From the all-important business model done in RASP with tool-support MacRASP, we export the general concept model to NIAM and start with detailed analysis of every information object.

In fact, most of the infological constraints is captured in the NIAM model. However, different aspects of performance, efficiency etc. are not covered in NIAM. Furthermore it is quite impossible to dynamically test or simulate the NIAM model.

Advantages and Problems

The three methods are characterized by high formal standards and underlying mathematical foundations. Most of the other methods and tools on the market lack the firm foundations that I believe will be sine qua non if systems engineering is going to be a mature discipline.

EXAMPLES OF MODELS

Trainers and Support Systems

Education and training of senior commanders is normally extremely costly. Making the training realistic requires the presence of the entire combat organization, including personnel, vehicles, weapons, sensors and communications equipment.

This means that the number of participants considerably exceeds the number of commanders and staff members being trained. It also involves long and complex preparation, and limited opportunities to reproduce all possible combat situations realistically.

A battlefield simulator, e.g. TRACCS, can be used for training personnel categories in the ability to act and take decisions under conditions of extreme stress. The basic requirements is that the personnel being trained should be able to work under conditions which are as realistic as possible.

The commander and staff in training operates with his staff in the normal environment, i.e. command vehicle with complete field equipment. The exercise director, the game controller and a number of operators representing subordinate and equivalent units must be simulated. The trainees communicate with the game operators using their ordinary equipment, connected to a telecom unit which simulates interfaces, background noise, and calculate audibility.

The activities within a staff must be thoroughly understood. Hence a RASP model is convenient at this stage of development. It is also easy to realize that much basic functionality in a computerized training centre such as TRACCS or ESIM have much in common with a battlefield C2 support system, e.g. AUSTACCS.

All the orders given are entered into the center's computer system by the game operators. The game controller can influence the exercise from a special operator position.

The exercise director follows events on a separate monitor, or on a video image projected on a wall. Before each exercise, the director sets up a game plan. He uses a database to select such factors as the geographical exercise area, hostile activities, friendly resources, communication conditions, weather, wind, additional phenomenology and finally the timing of different events. Maps of the relevant exercise area are retrieved from a map achieve or created with relatively advanced GIS functionality, so called map generalization.

During the exercise the positions of friendly, allied, neutral and, of course, hostile forces are displayed on different screens. Resources are counted down as they are consumed. The operators receive alert signal in a number of situations. The entire sequence of events is recorded in the center's computer system, as well as all voice communication.

The exercise director can halt the exercise at any time to give comments on actions taken. He can also reverse the exercise back to a situation in which the commander or a staff member has made a wrong decision, and allow them to correct the error by issuing of a new order.

The game plan can, of course, be used more than once. Thus a number of suitable standard games of varying degrees of difficulty can be stored into the system.

The information objects in systems like TRACCS or AUSTACCS are of four different types:

- unit organization;
- equipment characteristics;
- environment characteristics;
- game scenario and data, i.e. order sequences, weather conditions etc.

All this information is complex and of the same intricate complexity as computer generated forces. Therefore, a tool is needed for information analysis, such as NIAM.

The basic data is entered into the system once for all types of units that will be used in TRACCS. The different units, for example an infantry brigade, will be broken down into the lowest level that is supposed to be handled in the game. This depends on the purpose of the simulation. Remember that all models and simulations are on a par with the problem it is supposed to solve or test. Sometimes the brigade must be broken down to the squad level. These subunits are then described how they are organized concerning personnel, weapons, sensors, ammunition, vehicles etc. The subunits will also be defined concerning possible speed on roads, cross-country, consummation and damage of different types of resources depending on activity, dead and wounded personnel relying on various types of combat.

It is, of course, possible to add to this data base new types of units or new types of unit organization as well as let the units be equipped with new types of equipment. The procedure of adding new information to the data base will be made by the end user of the system.

In order to get a total and consistent model of the concept of battle and all its aspects it is furthermore necessary to have a total business model, e.g. a RASP model.

In order to be able to build an efficient computer system, concurrent and distributed and compatible to the DIS model, a corresponding HCPN model will be of great support during the implementation.

There are a lot of similarities between trainers and support systems. Most of the information object are the same, most of the transactions similar etc. However, a trainer for e.g. a staff has to be constructed according to the principles and routines that are generally accepted or formalized in official regulations. A C2 support system can be tailored much more independenly.

Operational Research for Joint Campaign Level

The joint level for somewhat different problems. It is very easy to distribute all information with modern computer support, but there are several principle problems with such a system. To link air, land and naval information support systems and have a decision-aid for operations planning at joint level and for developing operations strategy on a broad scale one must handle:

- abilities to assess campaign outcome;
- force deployment and effectiveness strategy;
- development of operational doctrine.

The attrition rates of both own and opponent units, weapon systems and sensors must be included.

The planning before the actual commencement of an armed conflict, where the operations planners will assess the overall situation and determine the possible courses of actions.

All these requirements must be formally managed in a business model and transformed into a information model, and finally into a systems model.

The methodological chain described above supports this work and guarantees consistency and completeness. Furthermore, the traceability gives, when new requirements becomes actual, a very useful advantage.

CFG with CLP – True Intelligent Objects?

Some of the essential problems with CGF are the lack of these objects to adapt to new situations and reason with respect to new factors.

Constraint Logic Programming (CLP) has recently attracted much attention as a technology for solving complex optimization, simulation and scheduling problems, e.g. military logistics and planning. It combines constraint solving with high-level problem-oriented programming, resulting in high programmer productivity in solving such problems.

Uppsala University and CelsiusTech are engaged in a joint research and development project, INTSIM, with the perspective to model intelligent and independent objects (rule-based) and fuse the software engineering concepts of logic programming and parallel process communication.

The project will use a essentially distributed process implementation strategy and use the programming language Erlang (developed at Ellemtel in Sweden) that well suits the actual subject domain of concurrent real-time distributed simulation objects.

CONCLUSION

The three methods presented in this paper and the methodological chain that encompasses the use of them has proven very efficient during several projects.

Moreover, the formal and mathematical rigid foundations these methods build upon is promising. There are, of course, some drawbacks. The formalism is regarded by many people as cumbersome and an obstacle that hinders intuitive creativity.

These opinions are in my view incorrect. If the systems we build can be compared to other engineering artifacts of man, the basics of our work must be founded on sound science and technology. CAD and other fancy drawing tools for modeling purposes can't survive that test and those demands.

It is a nice possibility to represent the behavior and the rules of interacting autonomous objects if each object is represented in the system as a concurrent process, hence the object is defined by a concurrent program, but the modes and protocols of interaction between the objects need to be specified, and the internal reasoning of each object must somehow be captured. Since concurrent constraint languages provide both communication and logical primitives, they are well suited for attacking this problem.

By considering objects as processes, it is also possible to introduce other processes, behaviorally similar to objects, but actually representing other users, or environmental and phenomenological entities. The net result being a set of interacting concurrent processes, which are driven by events and plans. For example, in the simulation of troop manoeuvres, each troop is defined as a concurrent program which interacts with other troops, but which also synchronizes with the time process and reacts to orders.

The problem of how a user is to interact with the system. The client-server model lends itself naturally, such that each user is a client communicating with the simulation server. Furthermore, a multi-moded interaction is necessary, i.e. at any instance a certain mode of interaction is appropriate. E.g. a commander may be listening to a series of battle events, widely distributed in space, instead of looking at it, to better appreciate an overview of the situation. At another instance, a close-up 3D-view of a battlefield is more appropriate.

The problem of interaction is further increased by several users, in particular when the different users cooperate in the system. Several options exist in how to solve this, ranging from electronic mail communication to establishing two-way audio and video links between users.

In the current post-socialist world every day seems to bring new conditions (technically, politically and economically) for the military. Consequently the disciplines of modeling and simulation, operational research and systems assessments will be of continued growing relevance. Synthetic environments where it is possible to test new sensor and weapon systems, new doctrines and organizational principles, and the art to model these objects will be important and crucial. Therefore the most mature and formal rigid methods must be used; and these methods must be linked together into an efficient methodological chain.

MODELING THE LITTORAL OCEAN FOR MILITARY APPLICATIONS

Steven D. Haeger
Naval Oceanographic Office
Stennis Space Center, Mississippi

ABSTRACT

Describing the ocean environment for military applications in shallow as opposed to deep water requires not only a shift in technology but also a change in our management and planning perspectives. The purpose of this paper is to discuss some of the issues that the training community should consider early on in the planning stages for modeling and simulating the coastal ocean in order to estimate funding levels, computer resources, communication capabilities, etc. Although the Navy has enjoyed relative success in modeling deep water for mainly ASW applications, our capability to model the coastal ocean is very limited. In shallow water, there are more warfare communities to address and more environmental parameters to model, including currents, waves, surf, visibility, bioluminescence, sediment dynamics, and temperature/salinity (with related variables of sound speed, density, conductivity).

Defining how well we can model the complex littoral region is relative to exactly how much and what type of information we need to know for a specific application. Environmental prediction models running in "real-time mode" describe what is happening in the ocean at a specific site and time but often must be run at a central site, are more expensive to implement/validate/run, and have a greater chance of not passing validation. Models running in a "typical scenario mode" provide time-varying answers that are statistically correct over some time period but are not accurate for real-time; they do not have the severe restraints listed for the real-time models.

Other issues include research and development models versus "operational" models, intentional rejection of assimilation data for ocean models; and determination of *unfair* fight criteria for platforms that are not capable of receiving model output from central sites for use in tactical decision aids.

ABOUT THE AUTHOR

Steve Haeger is a senior oceanographer in the Science Technology Staff at the Naval Oceanographic Office, with eighteen years of experience in the measurement, analyses, and modeling of shallow-water ocean processes. He has a B.A. in Physics from Illinois Wesleyan University and an M.S. in Physical Oceanography from Florida Institute of Technology.

MODELING THE LITTORAL OCEAN FOR MILITARY APPLICATIONS

Steven D. Haeger
Naval Oceanographic Office
Stennis Space Center, Mississippi

INTRODUCTION

During the last decade, the Navy has developed and implemented ocean models for operational applications. These operational models are run at Navy land-based centers or on platforms at sea with adequate computing capabilities; nowcasts or predictions are used for near-term planning of force deployment or on-scene tactical decisions. Historically, the models mainly addressed environmental affects on deep water Anti-Submarine Warfare (ASW) and Under Sea Warfare (USW). With the recent changes in the world geopolitical situation, emphasis is now focused on understanding and describing the littoral ocean environment. The use of models in the littoral will provide planning and tactical support for Navy operations; there are also many applications for training and education such as mission rehearsal, weapons and equipment training, wargaming, and engineering studies. Specific examples include training a sonar operator on a mine hunting sonar simulator, computer simulation of an amphibious assault in a Distributed Interactive Simulation (DIS) mode, and the computer design of a landing craft using simulated waves, surf, and beach topography.

The shift from deep to shallow water requires not only a change in technology but also a change in our management and planning perspectives. Because of the difficulty in modeling the highly variable coastal ocean, it is important to carefully define requirements for modeling applications to properly estimate funding levels, computer resources, communications capabilities, and to ensure smooth integration of ocean models into larger systems such as DIS efforts. This paper will discuss some of the issues that should be considered during the planning stages of defining modeling and simulation requirements for training or education applications. Note that the traditional operational models are also important for certain training applications such as wargaming or on-scene exercises.

What is the Littoral Ocean?

The littoral, as commonly defined by the Navy, extends from the coastline 200 nmi seaward to 60 nmi inland. A general definition for shallow water in terms of ocean dynamics includes continental shelves, coastal bays, estuaries, and rivers (generally water depths less than about 200 meters). Thus, the littoral ocean is comprised of both deep and shallow water, depending on the basin. This is of significance to ocean modelers who are tasked to model littoral regions, because ocean dynamics and model characteristics are often different between deep and shallow water, as described in a later section. Although ever-increasing computer capabilities and model sophistication are eroding the gap between the physics and data requirements of shallow-water and deep-water models, differences will exist for years or decades, depending on the application. The emphasis of this paper will be on the shallow portions of the littoral oceans.

Potential Ocean Models of Importance to Military in Littoral

The term "potential" ocean models refers to models that address specific ocean parameters of importance to Navy operations and are generally available in a research and development mode; the models may or may not be available in an operational mode. Only statistical and dynamic models that produce time-varying answers (time-series) will be included here:

Ocean Thermal Models: for sound speed (acoustics) or density (buoyancy of submerged platforms)
Ocean Circulation Models: for currents, water elevation, temperature/salinity (sound speed, density)
Tide Models: for tide height/currents (often a simplified version of circulation model)
Ocean Wave Models: for wave/swell height, direction, period (seaward of surf zone)
Surf Models: for surf characteristics and longshore/rip currents in surf zone

Sediment Transport Models: for scour around bottom objects or resuspension (affecting visibility)
Optical Models: for vertical and horizontal visibility
Contaminant Fate Models: for oil and other water constituents
Biological Models: for rates of biofouling growth, bioluminescence

Acoustic and atmospheric models will not be included in this paper. Note that some models must be coupled to others such as a sediment transport model (which would be coupled to a wave and circulation model). The most important issue with respect to these potential models is that many existing models falling into each of the categories above may be useful for investigating R&D problems, but few are adequate for applied use for the military. It is emphasized that in many situations, the problem is not in the models per se, but in the extreme lack of data for understanding the physics of the local basin, calibration of empirical formulations, assimilation, and validation. Models address R&D issues more adequately than applied modes of model implementation for several reasons: R&D implementations often are tailored to investigate a "single process" of a larger problem, often are site specific, usually have a customized data set, and are set up and run by personnel of higher expertise.

Operational Ocean Models in Use by the Navy in the Littoral

The Navy has enjoyed relative success in modeling deep water mainly for Anti-Submarine Warfare (ASW) and Under Sea Warfare (USW) applications; however, our capability to model the coastal ocean in general is quite limited for operational implementations. Operational is defined here as a model that is set up for a basin (example Yellow Sea), runs in a semi-automated mode (usually daily) using input data that is available on a daily basis, and provides near real-time or predictive products of tactical significance to the Fleet. The models that are currently being implemented in coastal areas by the Navy include Thermal, Circulation, Tide, Wave, and Surf. Most will provide accurate, meaningful answers in some basins, when sufficient data are available while running in a semi-automated mode; few will presently provide accurate answers under all types of oceanographic events in all regions of any given basin. This does not detract from their usefulness, it just means that we must be prudent in how and when the model outputs are used. As discussed in a later section, the validity of a specific model implementation is very dependent on the specific type of problem to be solved. Each model addresses multiple requirements for the Navy; some requirements or applications are scientifically easier to satisfy than others.

DIFFERENCES BETWEEN DEEP- AND SHALLOW-WATER MODELING

Spatial and Temporal Scales

One obvious difference between deep- and shallow-water modeling is the difference in oceanography; the scales of variability are usually much shorter in shallow than deep water. On continental shelves, spatial (horizontal) correlation-length scales for many processes are longer in the alongshelf direction than the cross-shelf direction, as depicted in the schematic in Figure 1. Common exceptions to this rule for continental shelves are local regions near tidal inlets, rivers, and canyons. Alongshelf and cross-shelf delineations become more confused within semi-enclosed seas such as the Adriatic or Yellow seas.

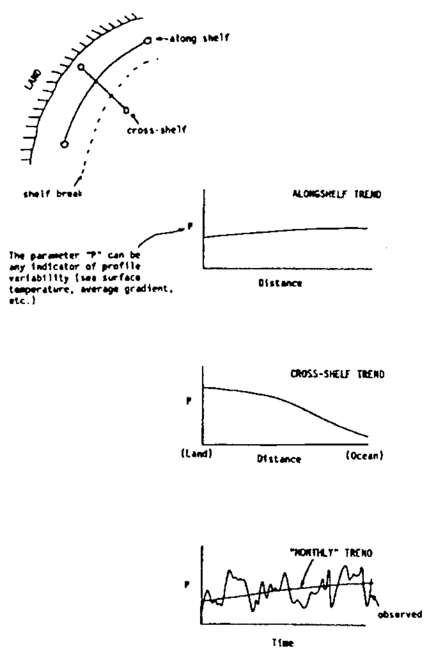


Figure 1. Spatial and Temporal Correlation-Length Scales on a Typical Continental Shelf

Temporal variability can be broken into seasonal and non-seasonal. For many oceanographic problems the expense of dynamic modeling is prohibitive, and the temporal variability of temperature, salinity, currents, etc., is captured in seasonal or monthly climatologies (mean values or statistics). However, in shallow water, the high frequency (non-seasonal) variability significantly impacts Naval weapon systems and operations. Note in the lower graph in Figure 1 that the arbitrary time series shows that the non-seasonal variability is greater than the monthly variability. Figure 2 is a time series of observed temperature at two depths on the continental shelf off Charleston, SC, at the 46-meter isobath from April through July. Note the high frequency variability compared to the longer term "seasonal" trend, especially in the temperature difference plot (temperature at 11 meters minus temperature at 43 meters). The non-seasonal variability is a result of many forcing mechanisms, including meteorological, tide, freshwater runoff, internal waves, and open-ocean processes impinging upon the shelf. In general, the magnitude of high frequency variability is greater on wide shelves than narrow ones; however, there are many exceptions to the rule.

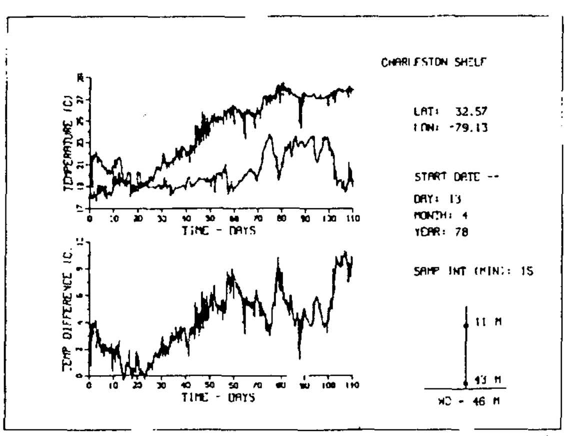


Figure 2. Ocean Temperature Time Series (11 and 43 meters) in Shallow Water.

Warfare Communities/Ocean Parameters

Naval operations in shallow water emphasize different warfare communities and different ocean parameters than in deep water. In past decades, the majority of ocean modeling was in support of ASW/USW acoustics. The important feature of the water column was the sound speed profile; other important environmental factors included wind/waves, bottom characteristics, water depth/bottom slope, ice, and noise from various sources. For Naval operations in littoral regions, more warfare communities are involved, and additional ocean parameters are of importance. Littoral warfare includes Mine, Special, Amphibious, and Strike Warfare as well as ASW/USW. A specific environmental factor such as temperature will affect the platforms and weapons systems in these communities differently, giving rise to the unpleasant issue that model validation will now require multiple criteria. A temperature (sound speed) profile produced by a model that is accurate enough for ASW acoustics may not be adequate for mine hunting acoustics.

The additional warfare communities require knowledge of environmental parameters never needed or emphasized by ASW/USW such as currents, surf, visibility, and sediment transport/suspension. Additionally, those parameters that were emphasized for deep water such as temperature/salinity now have multiple effects on Naval operations, even within a single community. For example, in Mine Warfare operations, temperature/salinity is not only emphasized for sound speed applications, but for density and conductivity considerations. The sensitivity of sound speed, density, and conductivity on temperature/salinity is all different; thus, a model that predicts temperature accurately enough for sound speed applications may not be adequate for density applications, where salinity is relatively more important than for sound speed.

Table 1 is an example of the various Naval operations affected by currents for Mine Warfare. Note that for different applications, a different modeling approach may be required or sufficient. For example, diver operations require real-time or predicted "instantaneous" currents at specific times. On the other hand, for subsequent mine burial, it usually doesn't matter if we know exactly when the mine is undergoing burial, but whether it buried from the time of suspected implant to the time of hunting/sweeping operations. Thus, statistics are adequate for the burial problem but not fully adequate for the diver problem.

Satellite Data

Use of Satellite Data in Deep Water. The use of satellite-derived data will be different in shallow water modeling than it has been in deep water. Satellite-derived thermal data have historically been very useful for tactical ASW applications in deep water. The data have been used mainly in two ways: to locate positions of large mesoscale ocean fronts and to assimilate directly into models in regions away from fronts. The frontal locations have been used to drive thermal analysis and dynamic models by the Naval Oceanographic Office (NAVOCEANO), Fleet Numerical Meteorology and Oceanography Center, Naval Research Laboratory, various Fleet oceanographic centers, and on-board tactical decision aids. The surface depiction of these fronts and associated eddies is useful because (1) consistent correlations exist between the surface and subsurface characteristics of the mesoscale features and (2) we have collected enough *in-situ* data to enable us to quantify the correlations. Shallow water presents a different situation.

Shelf Break Fronts. Fronts in many shallow-water regions of interest to the Navy can be divided into two types. The first is shelf-break fronts, which are usually "locked" into the bottom along some narrow isobath range. Examples of these types of fronts are the Shelf-Slope front (U.S. east coast north of Cape Hatteras), the Florida Current-Gulfstream (south of Cape Hatteras), and the Kuroshio (along the shelf edge in the East China Sea). Although the dynamics of these types of fronts are complex, the major characteristics of the fronts (slope, horizontal gradient, position of intersection with bottom) usually remain somewhat consistent over time periods of days or weeks and can thus be described statistically, given enough data. Fortunately, the aforementioned areas are heavily sampled, and the basic statistics of the fronts are known. For these types of fronts, satellite imagery can be used in much the same way as has been done for deep water. Fronts on the continental shelf proper are more troublesome.

Continental Shelf Fronts. Several types of fronts are combined for convenience here to define the second class of fronts. These include tidal plumes; river runoff plumes; and less well-structured, but more prevalent, "coastal fronts," which can include anything from weak horizontal gradients to coastal jets. These fronts are often within 40 km of the coast but can occur anywhere on the continental shelf, depending on the basin. They are a direct result of freshwater runoff, but their characteristics are highly variable. Tidal or strong river plumes may have horizontal salinity differences of a few to tens of ppt (parts per thousand); temperature differences may be large or insignificant. Coastal jets may also have salinity differences of tens of ppt in extreme cases (such as off the east coast of Nicaragua) but more often have differences of a few ppt or less. Again, temperature differences may be large or small. The main difference between these fronts on the continental shelf and their deep-water counterparts is that the subsurface structure, and even existence, of the front is highly variable in time; thus a statistical representation of the frontal characteristics often cannot be inferred from a surface depiction from satellite or other remotely sensed data. Temperature-salinity (TS) relationships are also highly variable near these frontal regions; thus infrared imagery is of little use quantitatively for salinity purposes.

Figure 3 is a composite of frontal positions in the Yellow Sea/East China Sea as derived by NAVOCEANO from Advanced Very High Resolution Radiometer (AVHRR) imagery from 1990 to 1993. The Kuroshio hugs the shelf

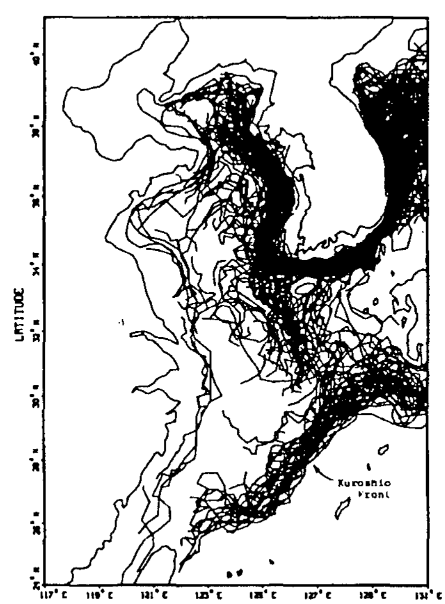


Figure 3. Ocean Temperature Frontal Locations in the Yellow Sea and East China Sea

break along the East China Sea; its subsurface structure is fairly well known as compared to the coastal fronts encircling the Yellow Sea. Are these coastal fronts tactically significant to Naval operations? Some probably are; however, much *in-situ* data correlated to satellite imagery will be required to determine surface to subsurface correlations. With enough data, we can possibly derive "mixed" correlations. For example, a weak thermal surface gradient may be an indicator of a strong subsurface turbidity (visibility) gradient during certain seasons or events. The future will show rapid advances in our ability to incorporate image-derived frontal information in shallow water; however, ocean models in use today have little capability to assimilate frontal characteristics in shallow water.

Assimilation of Satellite Data Away from Fronts. Satellite thermal imagery has also been used extensively in deep-water statistical models to infer subsurface temperature structure *away* from fronts (within "pure water masses") with considerable success. This has been possible for the same reasons we could infer frontal characteristics: the vertical correlations exist and there is enough data to establish the existence of and quantify the correlations. There is much evidence that shows that these vertical correlations often don't exist in shallow water except during isothermal conditions (constant temperature surface to bottom). The temperature difference plot in Figure 2 emphasizes this problem. Although the uppermost temperature sensor was at 11 meters and not the surface, the data imply that if sea surface temperature was known from imagery, little information could be inferred near the bottom. This is in contrast to deep water, where temperature variance decreases below the main thermocline. This is not to say that satellite imagery is not useful for shallow water modeling; only that we must rely less on statistical models in shallow water and more on full-physics dynamic models. The problem is that dynamic models require more implementation/validation effort, more computer resources, and a higher level of expertise to run, interpret, and calibrate.

Data Collection in Shallow Water

Our ability to model deep-water regions of tactical interest to the Navy is the result of having been able to make *in situ* measurements for several decades by research, merchant, and military vessels. Littoral regions off most third-world countries are data sparse; many of these regions are now off-limits to U.S. platforms, either completely or at least for the purposes of collecting oceanographic data. The problem is especially enhanced for those regions with narrow continental shelves, which may lie completely within the 12-mile limit. Also, the limited available historical data are often older and of questionable quality or low resolution. Remotely sensed data will tell us much qualitative information but cannot replace *in-situ* data as explained in the section above.

MODELING REQUIREMENTS

Defining how well we can model the complex littoral ocean is relative to exactly how much and what kind of information we need to know for a specific application. Presently, and in the near past, the Navy has emphasized the use of ocean models for near real-time or predictive applications, defined here as "operational." This mode of model application is the most demanding type in regard to model sophistication, data requirements, validation issues, computer resources, and sometimes modeler expertise. Other modes for training or education applications include mission planning and rehearsal, weapons and equipment training, and engineering studies. These other modes often do not require real-time or predicted answers, only representative answers over a period of simulated time. Even in those situations where the application will be for operational use, initial testing of the entire training system can usually be done in a non-real-time/predictive mode. The correct specification of the modeling mode may make the difference of whether an adequate model even exists or can be validated for a simulation. For example, a shallow-water circulation model that predicts tidal currents for a specific embayment may produce accurate amplitudes of the tidal signal (current speeds) but may easily be in error of one hour in phase. This can translate into significant errors in both speed and direction of "instantaneous" currents. A model with these errors would be adequate for many uses but not a real-time or predictive mode. Planners of training systems should decide whether a time series with proper statistics is adequate for the problem, as opposed to real-time/predictive answers, in order to estimate properly system development time, cost, data requirements, and computer resources.

Another decision to be made in modeling and simulation system planning is whether the full model output is required or only a product derived from the model output. In some cases with Navy operational modeling (3-D statistical thermal models), the model output is disseminated directly to the Fleet customer; it is somewhat out of the hands of the producer of the model grid as to how the customer might use the information. With other models (3-D circulation), the customer receives only a product that is derived off of the model output instead of receiving the entire 3-D grid. This screening process is used because certain models produce many types of answers, some being more accurate or verifiable than others. In defining modeling requirements for training applications, one should specify the specific application(s) required such as those listed in the example in Table 1. (More modeling requirements for other ocean parameters and warfare communities can be found in Haeger, 1993.) Obviously, it would initially simplify things if the customer blindly specified all possible applications for a specific ocean model; specific products could be created on the user end. However, as with specifying the modeling mode discussed above, overspecification of the model applications will significantly increase development time, cost, data requirements, and computer resources. Additionally, communications loads and data storage requirements on the user end will be greatly increased in order to handle full 3-D, time-varying grids.

OTHER ISSUES

Consistency of Modeled Fields

It is generally agreed that it is desirable to obtain consistency between different modeled ocean parameters for environmental simulations. For example, during a warfare exercise, sound speed profiles should not be obtained from a climatological data base while ocean currents are generated from a real-time model; the two dynamically related fields will be inconsistent with each other. This makes sense for platforms or weapons systems that are mutually affected by the same environmental factors in the same local region of the ocean. However, there are situations where inconsistent modeled fields may be useful. If the purpose of a simulation is to stress all weapons systems with extreme environmental conditions, then models and data bases may be used in an uncoupled manner. For example, a specific wind field that produces a water-column structure of extreme sound speed profiles for ASW acoustic performance may not be the same wind field that produces extreme surf heights that affect beach landing operations. It will obviously be more technically feasible in the near term to run models in a non-coupled fashion and not worry about consistency; thus, those scenarios in which these simulations can be useful should be identified as lower cost efforts.

Determination of Unfair Fight Criteria

In DIS applications, the fair fight issue necessitates the generation or dissemination of "equal environments" to all nodes. However, there are situations in ocean modeling where an *unfair* fight with regard to the environment should be built into the system. Naval platforms at sea obtain environmental information either by direct observation, by receipt of modeled fields/observations from a nearby platform or land-based center, or by modeled output generated by an on-board system. The various platforms have different capabilities to obtain this information; thus, an unfair fight is inherently built into actual operations. In other words, some platforms are capable of knowing the environment in time to use the information in tactical decision aids; others only feel the effect of the environment without having the ability to make tactical use of it. DIS architecture should have the capability to define appropriately who gets what information and when. Distinctions should also be made between modeled fields that can be observed on-scene (usually visually) by all platforms to some degree (wave height) and modeled fields that cannot be observed directly (sound speed). The platform that is not capable of receiving any modeled fields suffers more from not receiving sound speed fields than wave fields, since local waves can be visually measured to some extent.

Intentional Rejection of Assimilation Data

Statistical ocean thermal models assimilate observed temperature profile data to produce 3-D nowcasts of ocean temperature. If observed data are sparse relative to the pre-determined correlation length scales for that data, then assimilation of the data can actually produce results that degrade the big-picture description of the ocean. This problem is caused by our inability to know correct length scales for all regions of the coastal ocean and the fact that length scales in shallow water are event (time) dependent. For example, if a single temperature profile is collected by a mine hunter in a nearshore region, it may be better for the DIS to reject that profile for assimilation into a model rather than have its properties extrapolated to deeper water and affect ASW acoustics. The mine hunter will now obtain a different answer from the DIS-generated environment than from the observation it directly measured.

SUMMARY

Unrealistic specifications of littoral ocean modeling requirements for training and education applications will

APPLICATIONS	PARAMETER	TYPE OF ANSWER
Diver Operations	Speed, Direction	Instantaneous surface, subsurface
Mine Burial (Subsequent)	Speed	Subsurface Statistics (recent or predicted)
Underwater Vehicles (tethered)	Speed, Direction	Instantaneous Vertical Profile
Underwater Vehicles (untethered)	Speed, Direction	Instantaneous surface, subsurface
Drifting Mine	Speed, Direction	Time Averaged surface, (near-surface)
Navigation and Sweep Gear Offset	Speed, Direction	Instantaneous surface, (subsurface)
Sonar (variable depth) Offset, Trailback, and Oscillation	Speed, Direction	Instantaneous subsurface
Moored Mine Dip	Speed	Instantaneous subsurface
Lane marker Buoy Submergence	Speed	Instantaneous subsurface (surface)
Mine Roll, Walk	Speed, (Direction)	Subsurface statistics (recent or predicted)

These applications refer to "traditional" Mine Warfare operations which take place seaward of the surf zone.

Items in parentheses are of secondary importance.

Table 1. Mine Warfare Applications Affected by Currents

result in excessive developmental time, cost, computer resources, modeler expertise, communications load, and historical and real-time data requirements. Additionally, there is an increased risk that models cannot pass validation. Below is a summary of questions that planners should consider when designing training systems for various applications.

1) What mode will the model be used for? (Operational exercise, mission planning/rehearsal, equipment training, engineering studies). The answer determines whether the model must produce real-time/predicted time series, representative time series for a particular month/season, time series for specific littoral regions, or time series for generic littoral regions.

2) What are the specific warfare communities and tactical applications of the simulations? Do the applications require full-model output (time series of 2-D or 3-D grids), or will a simpler product derived from the model that is tailored for one or a few applications suffice?

3) Will the model be used in a "hands-off" mode, such as the present "approved" Navy operational models, or can R&D models be used? Is there a mechanism for a person of appropriate expertise to be in the loop to correct model output or tweak the model once simulations begin?

4) Can different modeled ocean parameters be inconsistent with each other?

5) Should an unfair fight be purposely designed into the system?

6) What is the protocol for platforms that observe data which disagree with "central site" modeled output? Can local platforms use observed data instead of modeled data?

These issues no doubt have some applicability to the modeling of other environments in the atmosphere, space, or on land.

REFERENCES

1. S. H. Haeger, "Naval Applications for Operational Ocean Models - Draft," in Coastal Ocean Modeling Workshop, (Sponsored by Office of Naval Research), University of Southern Mississippi, TR-2/93, June 1993

Large DIS Exercises - 100 Entities Out Of 100,000

Steven D. Swaine and Matthew A. Stapf
McDonnell Douglas Training Systems
McDonnell Douglas Aerospace
St. Louis, Missouri

ABSTRACT

Distributed Interactive Simulation (DIS) is being promoted as a tool to aid in design, prototyping and manufacturing of weapon systems, development of joint doctrine, mission planning, after-mission reviews and historical analysis. Future networked exercises promise hundreds of thousands of users interacting in a single "seamless battlefield". To achieve this goal, the evolving DIS standards have addressed the protocol for the data which must be exchanged between participants, and they embody a data reduction method known as "dead reckoning" to help reduce network bandwidth utilization. But this is only part of the solution. Connecting a simulator to a large exercise has been likened to "drinking from a firehose"; and most simulations, especially legacy simulators, simply cannot process the envisioned number of external entities. This paper discusses techniques for managing large quantities of entities, by filtering, organizing and prioritizing the DIS data for presentation to the simulation host.

ABOUT THE AUTHORS

Steven D. Swaine received his M.S. degree in Electrical Engineering from Washington University, St. Louis, Missouri in 1993 and a B.S. in Electrical Engineering from the University of Missouri at Rolla in 1986. His primary responsibility is in the area of research and development related to low cost simulation systems, particularly in the areas of microprocessor systems and communication protocols. He has served as the Principal Investigator for the MASS program which pioneered low-cost networked aircraft simulations and is the chief developer of the MDTS DIS Interface Unit. He also has been involved in the development of the DIS network standard.

Matthew A. Stapf received a B.S. in Computer Engineering from the University of Illinois Urbana-Champaign in 1985. After graduation, he came to work for McDonnell Douglas Flight Simulation Laboratory where he was a systems engineer responsible for the system design, installation, and maintenance of high-speed ethernet networks to support real-time traffic. Mr. Stapf is currently the Principal Investigator for the McDonnell Douglas Training Systems Simulation Networking research program.

Large DIS Exercises - 100 Entities Out Of 100,000

Steven D. Swaine and Matthew A. Stapf
McDonnell Douglas Training Systems
St. Louis, Missouri

INTRODUCTION

Most problems associated with large-scale simulation networking can be divided into two categories. The first category defines the minimal data set needed to communicate between nodes, its format and delivery mechanisms. These problems are addressed by application protocols, physical network architectures, data compression and communication services. They have been the primary focus of the DIS community to date and have resulted in the Standard for Distributed Interactive Simulation -- Application Protocols [1] and related documents. The primary concern for large exercises has been the amount of network bandwidth required. The DIS standard has employed a mechanism for bandwidth reduction known as "dead reckoning", which is essentially a lossy compression algorithm relying on statistical multiplexing. In addition, it is generally assumed that technical advances in communications hardware will eventually provide enough bandwidth for the very large DIS exercises envisioned in the future.

The second category, and focus of this paper, relates to data processing at the receiving nodes. These problems are addressed by network interfaces and involve data filtering, prioritization and organization. Computational technology is advancing quickly, but at a slower rate than communications technology, and most workstations and operating systems are not optimized to handle the data rates associated with large DIS exercises. In fact, the packet arrival rate usually becomes a problem for most DIS implementers today long before the available network bandwidth is fully utilized.

We begin by considering the problems associated with designing an interface for large exercises and then outline some generic mechanisms which aid in reducing the amount of network data presented to the host. Next, a specific implementation of a DIS network interface which utilizes many of these mechanisms is described. And finally, an example implementation is outlined detailing how a legacy fighter simulator could be interfaced to a DIS network.

INTEROPERABILITY CONSIDERATIONS

What does it mean to say that a simulation can interoperate with 100,000 network-supplied entities? Stated as such, it generally means very little. Perhaps the 100,000 entities are static bridges and the simulation is a submarine trainer. The trainer probably has no mechanisms for handling bridges and therefore expends very few resources in processing them. If the external entities are all enemy submarines located nearby, it is likely that resources will not be available to process all the entities and some portion will have to be rejected. If some are rejected can we still say the simulation is interoperable? Maybe, if entities are rejected that would be unavailable to (or rejected by) all the actual sensors then, as far as the operator is concerned, the situation is perceived as it would be in the real world and we have perceptual interoperability. If some of the information is not presented to an operator that should be, the simulator may still be partially interoperable. For example, if a tank operator sees 1000 enemy tanks come over a hill instead of 1001 the reaction of the operator may not differ, so we can still claim reactional interoperability. Of course there are other factors which must be included in judging host/exercise interoperability, such as the accuracy of the models used and the fidelity of the sensor representations (including visual). But, this paper focuses on the network interfaces and here we are concerned primarily with the processing of network data for the host system.

From the standpoint of the network interface, interoperability is less qualitative. Each host must dictate to the interface what network information can be rejected, or filtered while retaining the required level of interoperability. In addition to filtering out data not needed by the host, the interface may also have to prioritize the data to support a lower-level of interoperability during overload conditions. It is useful then to characterize the capabilities of an interface based upon its ability to process host requests and its ability to process network data (in terms of the network bandwidth, packet arrival rate and total numbers of entities). These two groups of metrics provide an insight into the extent of interoperability of an interface and can be used to compare the capabilities of various interfaces.

Note that network utilization and the packet arrival rate are not merely functions of the quantity of entities in a given DIS exercise. The characterization of a DIS exercise must also include information about the dead-reckoning algorithms, error thresholds and minimum send times used as well as the mix of entity types and their associated behaviors. Some work has been done in simulating DIS networks themselves to determine parameters of interest such as maximum transport delay, network utilization, packet arrival rates, etc. for a given network topology based on these exercise characterizations [2]. Thus, for a given mix of entities with known behaviors and a given network architecture, it is possible to characterize an exercise and precisely define the requirements for a given interface and a given host. However, the real challenge is to design a generic interface which can support a wide variety of hosts and can be used in broad range of exercises.

DESIGNING A INTERFACE FOR LARGE EXERCISES

The primary role of the network interface is to off-load the overhead of DIS related processing and input/output, and to regulate, under dynamic host control, the amount of data presented to the host. In addition, the network interface may perform other services such as data conversion, radio modeling, etc.. Since, for very large exercises, there will generally be many more external entities than internal entities, the network interface must focus primarily on the received data.

Most simulations have a limited amount of resources dedicated to processing external entity information; fortunately, most simulations also have finite perception capabilities. The process of removing data which is not of interest to the host is referred to as filtering or clipping. If the host is operating in an environment which exceeds its external entity processing capabilities, the interface must use prioritization techniques to decide what data should be returned to the host.

Filtering, a special case of prioritization (priority of zero), is best done as soon in the receiving process as possible to eliminate data from consideration with the minimal impact on system resources. For this reason, a filtering hierarchy, similar to a computer memory hierarchy, is often employed. It is important to note that the data movement from the low-level interface must be kept to an absolute minimum or else the time spent moving the data will dominate the reception process. This is a crippling limitation of many DIS implementations on workstations, which often must read an entire received packet into virtual space even if the packet is then immediately rejected.

Prioritization involves multiplying weighting factors by a set of entity parameters and adding together the results to achieve an overall priority index for each entity. These parameters include, but are not limited to, entity type, relative location and velocity, and allegiance. For example, a fighter pilot will be much more interested in a enemy aircraft 10 miles away headed towards him and within his field-of-view than in a friendly guided missile targeting a site 500 miles away.

In the next few paragraphs, we consider the transmission of Protocol Data Units (PDUs) from an issuing simulation as they pass through a network and finally are made available to the receiving host processor. Figure 1 summarizes these concepts. Not detailed, but pervasive throughout the hierarchy, will also be numerous checks to attempt to eliminate erroneous data using various hardware and software mechanisms.

Send-Time and Network Operations

Send-Time Filtering. The first opportunity to eliminate a PDU from the data stream is in the originator of the PDU itself. One example is the proposed aggregation/deaggregation protocol. Under this protocol, a simulator may represent a group of entities as a single aggregate entity until deaggregation is requested. Dead reckoning itself is another example of send filtering. Adaptive network loading has been demonstrated as a technique which involves dynamically varying the dead reckoning error thresholds as a function network utilization.

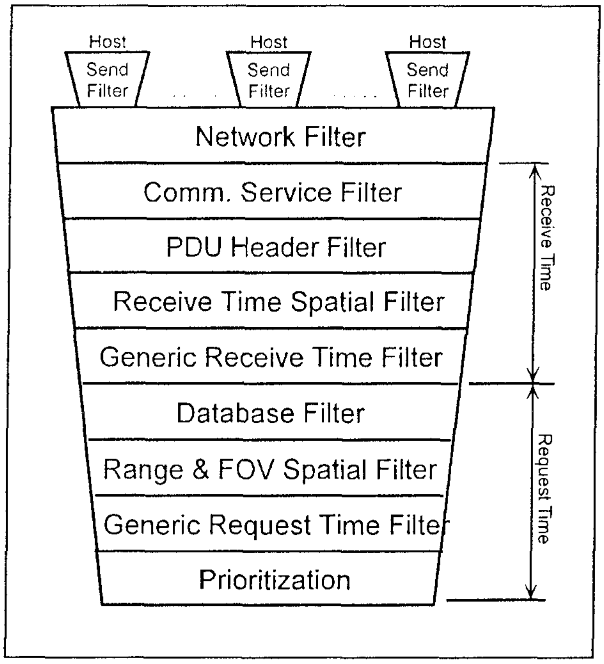


Figure 1. Filter Architecture

Network Filtering. Network filtering utilizes intelligent routers to prevent data from passing to sections of the network where it is not needed. Most research in this area involves the use of dynamic multicast addressing and battlefield gridding. Each grid square in the gaming area is assigned a unique multicast address which is used as the destination address in PDUs issued from entities contained within that grid square. Each host configures its network interface to only allow accept PDUs from entities in grids within its range of perception. This signaling information is then propagated out from the receiver to intelligent routers.

Note that send filtering and network filtering have the added advantage of reducing network utilization, since data is not passed to sections of the network where it is not required. One drawback to network filtering is that entities with large ranges of perception, such as threat environments, environmental servers, data loggers, network monitors and plan view displays present obvious problems.

Receive Time Operations

Communication Service Filtering. Communication service filtering generally falls in the Data Link, Network and Transport Layers of the OSI/ISO reference model. The first opportunity for filtering is examination of the network address. With ethernet this would imply a simple examination of the destination address of an incoming packet to check for a multicast or individually addressed packet meant for the host system. The advantage of these filters are that they generally are implemented in hardware and require no run time software cost. In general, these filters eliminate any non-DIS data which shares the physical network media. Other examples of communication service filters which exist further up in the Network and Transport Layers include the Internet Protocol (IP) type and IP address and the User Datagram Protocol (UDP) port number. These filters also help eliminate some of the non-DIS traffic which may be on the physical network. If additional information, such as exercise identifiers, PDU protocol family or originator location can be encoded into the port number then PDUs can be eliminated without any interface processing. Even further, if this information were encoded into the network address (via multicast addressing) then the PDU can be eliminated again at the hardware level. Data which passes through the communication services filters should then be DIS PDUs which must be processed to determine their value to the host simulation.

Protocol (or PDU Header) Filtering. If not eliminated by any of the communication services filters, the header of the DIS PDU can be examined for relevance. Some basic fields would be the exercise ID, protocol family and protocol version. Note that some interfaces may support multiple DIS versions.

Receive-time Spatial Filtering. Receive-time spatial filtering involves the removal of entities that the host simulation could not detect by any means due to the physical geometry between the two. To determine the relative geometry required between the two entities for perception, the host simulation must define its perception volume. The host's perception volume is often defined by a Center-Of-Interest (COI) and the maximum range of all the sensors on the receiving system (visual, electromagnetic, radar, etc.). Usually, the COI will be the location of the host entity; however, on some systems, such as Semi-Automated Force (SAF) systems, the perception volume is not always so easily defined. On these systems an entity centroid or the entire gaming area could be specified so as to accept entities only within a broad geographical area.

It is tempting to eliminate Entity State PDUs representing entities which are currently outside the Field-Of-View (FOV) of any host sensors. However, due to DIS dead reckoning, the network interface must maintain entity information covering a much larger volume than the immediate FOV of the sensors (Figure 2). For example, if an Entity State PDU is received for an entity located behind the observer, this PDU should not be filtered because the observer can turn and look behind him in less than the DIS minimum send time. In addition, the rejection ranges must take into consideration the relative velocities of the entities and actually be larger than the fixed perception range. Equation 1 details this relationship.

$$R_r = R_{smax} + (V_{coimax} + V_{emax}) * (T_{smin} + T_{fconfig})$$

R_r -	Rejection range, all entities with greater ranges are rejected
R_{smax} -	Maximum range of any sensor (including visual)
V_{coimax} -	Maximum velocity of the COI
V_{emax} -	Maximum velocity of any external entity
T_{smin} -	Minimum send time for DIS Entity State PDUs
$T_{fconfig}$ -	Filter reconfiguration time

Equation 1. Rejection Range

Other Receive-Time Filtering. It is impossible to anticipate all the combinations of filtering operations that may be desired by all applications; therefore, most interfaces include some mechanism to allow the host to specify filters in a generic manner. One such mechanism is for the network interface to perform a series of logical operations on the results of comparisons between user supplied data and fields within incoming PDUs. Using such generic filter specifications, the application could, for example, filter Transmitter PDUs that do not match the host's current Crypto Key ID

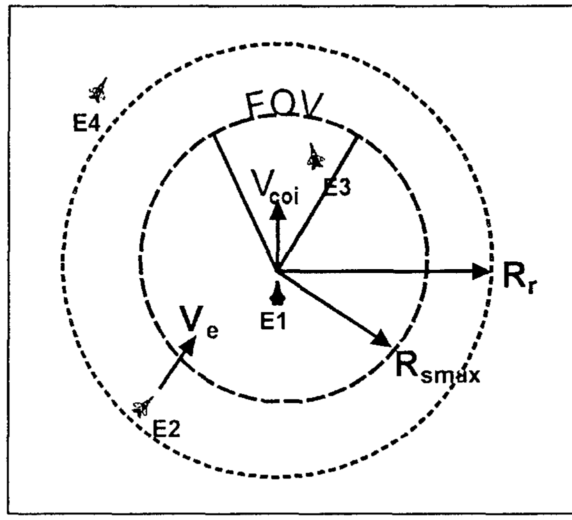


Figure 2. Rejection Range Illustration

Request-Time Operations

Data Organization. If a PDU is accepted by the network interface process, it must then be queued for processing by the host. It is often desirable to logically group data by PDU type or family so that independent host processes can address certain data types. The same generic mechanism used for receive-time filtering can also be used to partition the external entity database into logical groupings or "views". Later, when information is requested from the interface, it can be requested from one of these preconfigured views, greatly reducing the request processing time that would result from scanning the entire database for specific entity sets.

Request-Time Spatial Filtering. As stated previously, the receive-time range rejection is usually best done using a spherical or rectangular volume. However, the volume specified at request-time may be tailored to the instance of the request. For example, the volume may be defined in terms of the FOV of a sensor, or even within a more generic polygonal bounding volume pre-defined by the host. In addition to removing entities not within the field-of-view and performing range rejection, other fidelity/processing decisions can be made at this point. For example, perhaps only the very nearest entities require smoothing, and entities just within the rejection range may not require rotational dead reckoning. Figure 3 illustrates this concept.

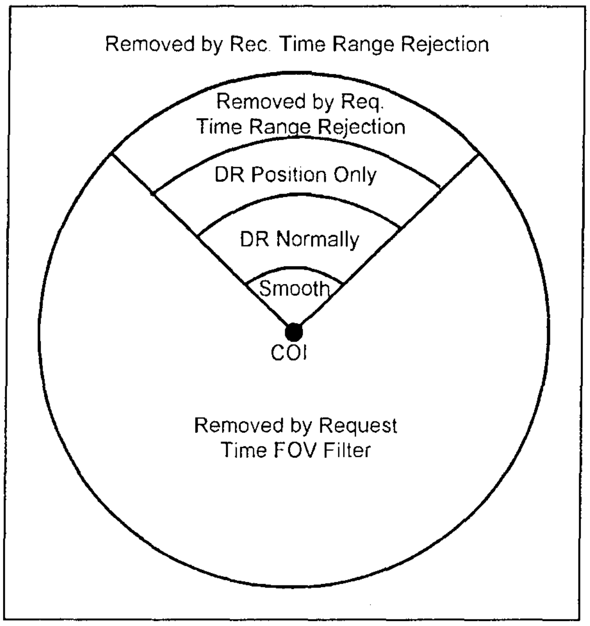


Figure 3. Range Based Fidelity Considerations

A simple dot product can be used to determine if an external entity is within a conical FOV. In the simplest case, a partial dot product can be used to eliminate entities located behind the observer (FOV=180°). The following equations illustrate the process of conical FOV filtering.

FOV = conical Field-Of-View
alpha = 1/cos2(FOV/2) (note, this can be calculated once, at initialization)
T = location of external entity
P = location of conical vertex (the ownship location is usually used)
N = Normalized axis of the conical volume (look angle or viewport)

$d_x = T_x - P_x$, $d_y = T_y - P_y$, $d_z = T_z - P_z$
$D = d_x * N_x + d_y * N_y + d_z * N_z$
if $D < 0$
then the target is behind the viewpoint, stop and reject here (done if FOV=180°)
if $D * D * alpha >= d_x * d_x + d_y * d_y + d_z * d_z$
then it is in the FOV, accept

Equation 2. FOV Inclusion Calculation

Other Request-Time Filtering. The same generic mechanisms for receive-time filtering can also be employed at request time. By doing so, the application has even more control in reducing the data that is presented to the host. This is especially useful when more than one application shares a network interface, because each application can have independent filter sets which are applied to a common entity database. Meaning, all the data on the network which is of interest to either of the applications which share the interface must pass the receive-time filter stage; however, separate request-time filtering can be imposed by each host.

Request-Time Prioritization. If, after all filtering is complete, there is still more data then the host can process, some data must be eliminated. This is most simply accomplished by truncation, but usually sorting based on range from the COI gives a more desirable result. Note that because sorting is at best a k log n algorithm (the closest k entities selected from a list of n) it is imperative that as many entities as possible are filtered before this final step. Other prioritization options include giving preference to enemy entities or to entities approaching the host's entity.

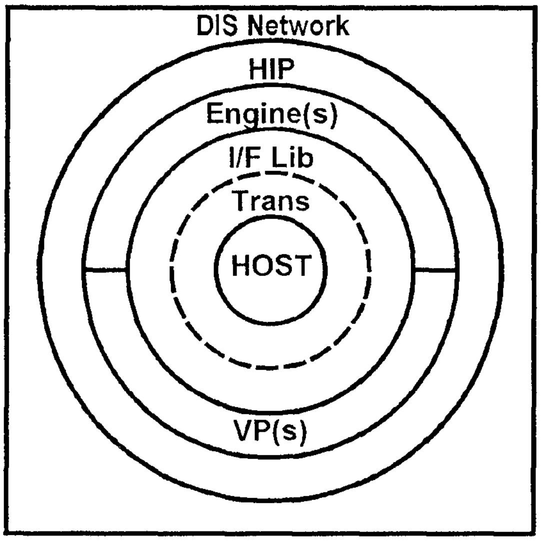
AN EXAMPLE IMPLEMENTATION OF A GENERIC DIS INTERFACE UNIT

We now outline a specific implementation of a DIS Interface Unit (DIU) which utilizes many of the mechanisms outlined thus far.

In the development of the DIU, speed was sacrificed only when the generic design or platform independence would have been compromised. In general, any complexities in the interface due to the emphasis on performance are abstracted out by the interface libraries into user-friendly calls. While C++ or Ada would have been well suited for this application, C was used due to portability and speed considerations. Interface libraries can be developed in any language, the preferable choice would be that of the host or intermediate translator process. Figure 4 illustrates the software architecture of the DIU. The communication mechanisms between the layers and the functions of each individual layer are described in the next sections.

Hardware Interface Package

The interface to the DIS network and all other platform specific interfaces are entirely encapsulated in the Hardware Interface Package (HIP) which allows the Engine, the heart of the DIU, to be completely independent of the hardware, operating system, and communication service. It is generally desirable to optimize the HIP for each platform, and it should be stressed that the performance of the entire DIU depends largely on the speed and capability of the HIP.



HIP - Hardware Interface Package
I/F Lib - Interface Library
TRANS - Translator Application
VP - Voice Processor

Figure 4. DIU Software Architecture

Many workstations are crippled in that the interface must move the entire packet from the physical network queue into user space even if it will be immediately filtered[1]. Note that if the HIP has access to the hardware, it is possible to dynamically re-allocate network queue buffers so that packets can even be accepted and filed into the interface with no data movement.

Another limitation of most workstations is the difficulty accessing high-resolution timers, and accurately determining the time of packet arrival is often completely impossible. In addition, with most operating systems it is not possible to determine the size of the packet on the network backbone, which compromises the network statistics gathering capabilities of the interface[2].

DIU Engine

The heart of the DIU is the Engine. It consists of several modules which are linked with a Hardware Interface Package to provide an executable DIU. The Executive Unit is the Engine's main loop. It controls operations within the Engine. The Receive Unit is responsible for processing incoming PDUs. It performs receive-time filtering, network variance compensation or transport delay compensation when absolute time is available. The Transmit Unit is responsible maintaining the extrapolated position/orientation of each host entity and sending state data only when necessary. The Statistics Unit accumulates information about network and DIU loading and events. This information is then available to the host or to a network monitor. The Command Unit is the primary application interface, handling all configuration and run-time data requests. It is responsible for entity extrapolation, smoothing and request-time filtering.

Engine Send Process. The DIU Engine maintains separate output "slots" for each internally simulated entity. This allows the Engine to accumulate statistics at the entity level, and allows the host to allocate PDU space within the Engine itself. The advantage of keeping the memory within the Engine is that the host is only required to fill the whole PDU once at initialization, then during real-time, only the dynamic fields need to be modified before instructing the Engine that the data is ready to

---

[1] Berkeley Standard Distribution (BSD) does support the capability to "peek" into a received message; however, the data must still be moved to a user buffer, it is just left in the queue for the next call to read again. To check the header information, the peek function could be used with a small buffer specified so that the entire packet would not be moved unless it passed the protocol (header) filters. This technique will not work for generic filters, however, since the test fields could be located anywhere within the PDU. One approach (when forced to use BSD) is to keep track of the percentage of PDUs being rejected based on header filters, and use the peek method only when it becomes more efficient.

[2] Silicon Graphics IRIX™ operating system does allow timestamp of received packets and size on backbone, but still must move all received data into user space. Silicon Graphics' raw "snoop" protocol allows the setting of "filters" but, unfortunately, do not support offsets greater than the size of a UDP header.

transmit. Considering that as much as 40 percent of an Entity State PDU is static or padding, this results in a significant time savings.

Engine Receive Process. After an incoming PDU passes the network filters, it will be loaded into the input queue by the network interface hardware. When possible, the HIP will also time-tag the packet at its reception. If present, the operating system will perform communication service filtering; otherwise, the HIP will perform these functions. Once this process is completed a pointer to the PDU itself is then made available to the Receive Unit on the Engine.

The Receive Unit then performs remaining receive-time filtering operations including removing PDUs with unrecognized DIS version fields and, if so directed by the application, PDUs with out-of-range exercise identifiers. Applications may specify unique receive-time generic filter objects to be applied by PDU type. The mechanism for the specification of these filters is further described later. If the PDU is an Entity State PDU, network variance or transport delay compensation can be performed and the entity accepted into the State Database. If this is the first PDU received for this entity, then a number of additional steps are performed. The entity is subjected to all the view filters which have been loaded by the application, and if it is found to be a member of a view, it is entered into the linked view list for that view. View zero is reserved and always represents the entire State Database. An entity may be a member of multiple views.

Views may also be specified as ranged-views. Ranged views are configured with a maximum number of entities or a maximum range from a given COI and the background task will constantly update the view membership of each entity. This further eliminates entities from consideration and since the background check is much faster then the DIS minimum send time, the range can be less then the receive-time rejection range.

It is the size of the State Database which dictates the number of external entities that the Engine maintains available for immediate request. On a 8MB MVME187 single-board computer, the State Database can be sized for over 10,000 external entities without using virtual memory. Note that this is not the same as the number of entities on the network since many entities will have been removed by receive-time filtering. Only the entities which might be of interest within the minimum send time need to be in the database.

If the PDU is a transient PDU (not an Entity State PDU), then it is optionally subjected to the database filter. Here, the PDU may be removed if it relates to an entity which is not currently in the State Database based on a fixed set of rules for each PDU type. For example, if the emitting ID in a Electromagnetic Emissions PDU is not found in the State Database, then the PDU is discarded, otherwise it is placed into a transient receive buffer. These buffers are also maintained on the Engine so that only the data actually used by the host need be moved from the Engine. There are two sets of double buffers, one for PDUs of the audio family and one for remaining PDUs. This allows the audio data stream to be processed separately and at a different rate than the remaining data. There is little advantage in further segregating the PDUs at this point, since the all PDUs not of interest to the host have been removed.

Interface Libraries

The interface library links with the host (or intermediate translator) application to provide a consistent API to the Engine. Interface libraries can be written in any language as required using object-oriented paradigms and extensive error checking. As previously stated, the primary role of the library is to abstract the complexities of the shared memory interface which was optimized for speed and efficiency and not ease-of-use. It also contains many utilities for allocating and filling PDU structures and building the generic filter specifications which are used for generic receive and request-time filters and in determining view membership.

A generic filter consists of one or more "terms", which specify a logical operation between a given value and a field within an incoming PDU. A term consists of a PDU field offset, a field type, an operation and a constant test data element. Each term also includes a global "and" or "or" operation which dictates how it logically relates to the previous term. The general form of the filter specification, selected for run-time efficiency, must match the following logical form:

(term [&& term ...]) [|| (term [&& term ...]) || ...]

where the brackets enclose optional terms and global operations. In words, a filter specification consists of one or more logical terms "and'd" and "or'd" together, with the "ands" taking precedence. Parenthesis are not supported, for run-time efficiency, but could be incorporated at the interface library level and removed before passing the data to the Engine. A syntax for generic filters has been developed and incorporated into the interface library which allows the application to easily specify filters in terms of ASCII strings. The Backus-Naur Form (BNF)[3] grammar of this syntax is shown in Figure 5.

<table>
<tr><td><filter></td><td>:= "<series>"</td></tr>
<tr><td><series></td><td>:= <spec> | <series>&&<spec>
| <series>||<spec></td></tr>
<tr><td><spec> :=</td><td><natural> <type> <oper>
<number></td></tr>
<tr><td><type> :=</td><td>c | s | l | uc | us | ul | f | d</td></tr>
<tr><td><oper> :=</td><td>== | != | < | > | <= | >=</td></tr>
<tr><td><number> :=</td><td>-<natural> | <natural>
| -natural>.<natural>
| <natural>.<natural></td></tr>
<tr><td><natural></td><td>:= <digit> | <natural>|<digit></td></tr>
<tr><td><digit> :=</td><td>0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9</td></tr>
</table>

Figure 5. Backus-Naur Form Specification

As an example, consider a filter object that will be loaded as the Collision PDU receive-time generic filter which will reject Collision PDUs when the colliding site is not 100 or the colliding application is not 200. The associated filter string is:

"18 us != 100 || 20 us != 200"

Which can be interpreted: "reject this packet if the unsigned short at offset 18 (the colliding site) is not equal to 100 or the unsigned short at offset 20 (the colliding application) is not equal to 200." It is sometimes the case that fewer terms are necessary, or filters can be processed faster, if a filter is specified in terms of when to accept a PDU rather then when to reject it. Both methods are supported by the Engine. The interface library also supports generic filter production utilities which allow the application to build and load filters without having to know the offsets of the fields.

ENGINE PERFORMANCE ON THE MVME197 SINGLE BOARD COMPUTER

We now consider the performance of the DIU Engine hosted on the Motorola MVME197. The MVME197 is a single-slot VME board available with either one or two 88110 processors, embedded ethernet, serial and SCSI interfaces. The dual processor board is particularly well suited for DIS applications since one processor can be dedicated to the Engine and the other processor can perform any translator and host interface functions. The MVME197's on-board ethernet interface was used as the DIS network interface for the following benchmarks. The Hardware Interface Package was developed on top of the bare board -- no operating system was used. For purposes of these timings, the host is considered to be the translator. Of course, if the host is further removed, say another processor in the VME or a separate computer with a VME interface, then these times would increase. When not explicitly stated otherwise, timing represents nominal times. To simplify comparisons, dead reckoning algorithm two (FPW) was used and Entity State PDUs had zero articulation parameters. Also, no data conversion or smoothing was requested of the Engine.

Send Statistics

Once the translator has loaded the PDU information into the Engine, a single call to an interface library routine instructs the Engine to issue the PDU. The Engine, in turn, then submits the PDU to the HIP which returns immediately. Because the data is never moved from the buffer, but instead is accessed directly by the ethernet hardware, the buffer is marked "in-use" until the packet has been issued.

Host time to pack data: (varies based on the amount of data updated in the PDU)
position, attitude and DR fields only: **10μsec**

Host time to command a send:
Engine time to pre-process the PDU and decide to send it: **60μsec**

Engine time to pre-process the PDU and decide not to send it: **25μsec**

Total time until the data area is again available on a lightly loaded network (packet send complete):	**225μsec**

Receive Statistics:

Interrupt Handler (per packet):	**18μsec**
HIP processing (per packet):	**9μsec**
Processing time for a PDU rejected by exercise ID:	**5μsec**
Processing time for a PDU rejected by third term of generic filter:	**9μsec**
Processing time for a accepted Entity State PDU:	**25μsec**
Processing time for a accepted transient PDU:	**20μsec**

Request Statistics
Again the focus is on Entity State PDUs related to entities which are currently in the requested view in the State Database.

Return 100 entities from a view of 100 (No sorting):	**3.1msec**
Return 100 entities from a view of 300 (Sorted and FOV filtering enabled):	**3.9msec**
Return 100 entities from a view of 1000 (Sorted and FOV filtering enabled)	**8.5msec**
Return 100 entities from a view of 5000 (Sorted and FOV filtering enabled):	**25.2msec**

For each of the previous cases where FOV filtering was used, the FOV filter first reduced the number of candidates to approximately one-third of the total in the view. The remaining entities were sorted to return the closest 100. Clipping plane filtering and request-time generic filters were not used in these measurements.

AN EXAMPLE IMPLEMENTATION TO A LEGACY SIMULATION

In this section we consider the problem of interfacing a legacy fighter simulation to a large DIS exercise using the DIU. The elementary frame rate of the simulator is 20 Hertz and all Engine calculations are completed within a single frame. The simulator has the following sensor restrictions due to the host simulator hardware, software and/or actual avionics system.

Air-to-air (A/A) radar - 100 air entities within 120 degrees FOV and out to a range of 80nm.

Visual system - 100 entities within a 45 degree FOV out to 15nm from the ownship.

Air-to-ground (A/G) radar - 100 land entities within a geometric volume approximating the scan width of the radar during a single frame out to a range of 40nm. This volume is a five degree "wedge" defined by two vertical planes intersecting at the fighter. By using this technique, thousands of entities can be displayed on the radar.

Based on this information, the maximum external entity density can then be calculated for each sensor. To simplify this analysis, a uniform entity distribution is assumed. A better method would be to use a non-uniform distribution (perhaps a Poisson distribution). The degree of interoperability is then characterized in a probabilistic fashion (i.e. the simulator has a certain probability that it will be fully interoperable in a given exercise). For density calculations, the perception volumes have been projected onto a plane and the area is calculated. This admittedly greatly reduces the number of calculated entities which the interface must maintain; however, most of the entities in a very large exercise will generally be ground-based or close to the ground. With ground-based entities it is reasonable to assume that there will not be more than one entity in any vertical projection. Even with air entities, it is usually more intuitive to convey density information in terms of area (i.e. 1000 aircraft within a 60nm by 60nm area) rather than in terms of a volume.

Entities Densities Used:

A/A radar -	0.015 Air Domain Entities/nm2
Visual -	1.132 Total Entities/nm2
A/G radar -	1.432 Gnd Domain Entities/nm2

Next, we calculate the maximum rejection range, which will determine the subset of entities accepted into the State Database (View 0). The maximum velocity of the ownship is assumed to be 1000 knots, and the maximum velocity of any entity of interest is 1000 knots. The network will have a minimum send time of thirty seconds and the receive-time range rejection filter reconfiguration time on the DIU

Engine is negligible. Therefore, based on the formula for receive-time range rejection (Equation 1), we must add an additional 17nm to the A/G radar range of 40nm to calculate the final receive-time rejection range for ground-based entities of 57nm from the ownship. Similarly, for air platforms, the rejection range is 97nm.

Next we define the rejection range for all of the views. Recall that the DIU Engine will check each received entity in the background for migration into and out of ranged views. The worst case time for this check is five seconds, which is the filter reconfiguration time. Since the data is immediately available in the database, the effective minimum send time is zero and the view rejection range must only be increased by 2.8nm in each view.

Using these results, the number of entities that the interface must support in each view can be calculated. For example, the air view must support 322 entities (.015 entities/nm^2 $\times$ π $\times$ 82.8 nm^2)(see Figure 6).

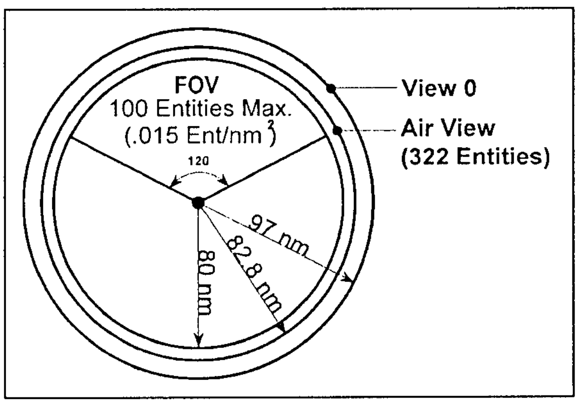


Figure 6. Air View

Calculated Maximum Entities Per View:

A/A radar	- 322
Visual	- 1127
A/G radar	- 8243

1) A/A Radar View
filter - accept only air domain platforms
COI - ownship position
range - 83nm

2) Visual View
filter - none
COI - ownship position
range - 18nm

3) A/G Radar View
filter - Accept only ground domain platforms
COI - ownship position
range - 43nm

Once per host cycle, request are then made as follows:

1) Request A/A radar information from the A/A radar view, sorting to return the closest 100 entities to the ownship within the radar's FOV.

2) Request the visual data from the visual view to return the closest 100 entities to the ownship within the visual FOV.

3) Request A/G radar information from the A/G radar view, using clipping planes to reject entities outside the current radar swath and return the first 100 entities encountered.

Processing Considerations

Recall that while the Engine is processing these requests, the host is free to perform other tasks, such as processing received transients or performing send operations. Therefore, these sequential requests actually represent the worst case path. Based on the performance data of the MVME197 processor board, it can be seen that the memory and processing capabilities exist to service hosts requests. The previous analysis shows that the fighter simulator would be fully interoperable and that the limiting factor is the host itself and not the network interface. If other sensors are required (such as FLIR), additional engines could be added in parallel to support them.

Turning attention to the receive process, since the visual view will include all ground-based entities, the limiting densities are .015 air platforms per square nautical mile over an area out to 97nm from the ownship and 1.132 total entities per square nautical mile over an area out to 57nm. This corresponds to about 12,000 entities which must be accepted into the State Database when operating in fully interoperable mode. If we assume the use of ethernet and UDP/IP and a minimum send time of 30 seconds, the data rate associated with 12,000 static entities with no articulated parts is much less than one megabit per second and the packet rate is only 400 packets per second. Of course, entities which send with a greater frequency, or entities with articulated parts and transient PDUs will greatly increase these numbers. If we assume one percent of the entities issue Entity State PDUs at a rate of five PDUs/second and ten percent of the entities issue Entity State PDUs at a rate of one PDU/second while the rest are static, then the total rate for the data which will be accepted by the interface is about 2156 packets/second.

Since 100,000 external static entities will generate over 30 megabits/second of Entity State data, ethernet's 10 megabits/second bandwidth will be exceeded unless some sort of network filtering is employed. One solution is to use dynamic multicast addressing and separate multicast address groups for air versus all other entities. If we define database grids which cover 60nm by 60nm at the ground level, the interface needs to subscribe to 16 air grids and four general entity grids (which also covers a five second network filter reconfiguration time). Based on the worst-case densities, this corresponds to a receive load of about 17,000 entities. Using the previous ratios for network loading, this corresponds to a packet rate of about 3085 packets/second.

Experimental Comparisons

In order to validate the load on the DIU Engine in the previous example, an actual prototype implementation of this exercise environment was constructed. A SAF system was used to generate the 17,000 entities with the appropriate distribution to simulate the input from the network into the ethernet interface. The following processing statistics were gathered by the DIU Engine in the fighter simulator.

65%	-Host Requests for Entities
22%	-Network Processing
<1%	-Statistics Gathering
<1%	-Sending of Host PDU's

<89%	Total Engine Processing Load

CONCLUSIONS

Most networked, man-in-the-loop simulations strive to achieve a high level of reactional interoperability. This level of interoperbility mandates that the user reacts to given simulated situations in the same manner as they would in the real world. For large exercises, the role of the network interface is to filter and organize inforation for the host without compromising the required level of interoperability. Further, the interface should prioritize the data for graceful degradation when operating in overload conditions. As has been illustrated, by analyzing the simulator capabilites and making some assumptions on external entity distribution, the limits on external entity desities can be determined. This information also dictates the requirements of the network interface.

On the basis of the Engine processing loads detailed in the previous experiment, it is possible to attach a multi-sensor legacy simulation which only supports approximately 100 entities per sensor to a network of 100,000. In order to interface such a simulation with ethernet networks and current processing technology, dynamic multicast addressing should be used in combination with extensive filtering and entity database management techniques. Such an interface does not require costly hardware or removal of entities which would normally be perceived in a real world scenario. It merely requires the application of prior knowledge to help the network interface anticipate what information the host will be requesting. This application of prior knowledge allows the network interface to reject information outright which will not be of interest to the host and therefore greatly reduces processing requirements placed on the network interface.

REFERENCES

1. Standard for Distributed Interactive Simulation -- Application Protocols, Version 2.0, Fourth Draft, IST-CR-93-40, Institute for Simulation and Training, 3280 Progress Drive, Orlando, FL, 32836.

2. Dille, J. and S. Swaine, *"Discrete Event Simulation and Analysis of DIS Network Architectures"*, Proceedings of the 14th Interservice/Industry Training Systems and Education Conference, 1992.

3. Pagan, Frank P., "Formal Specification of Programming Languages: A Panoramic Primer", Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1981.

A DIS NETWORK FOR EVALUATING TRAINING SYSTEMS EFFECTIVENESS

Christina L. Bouwens and Robert E. Jones
CAE-Link Corporation
and
Dr. Linda Pierce
Army Research Laboratory Ft. Sill Field Element
Ft. Sill, Oklahoma

ABSTRACT

Distributed Interactive Simulation (DIS) has recently received widespread acceptance in the DoD community as the standard for networking simulations. Although DIS has its roots in interfacing virtual simulations (entity level, typically man-in-the-loop training), it is also being adapted for use with constructive simulations (wargame and analysis) and live simulations (real, fielded equipment). This paper describes a DIS network combining live constructive and virtual simulations. The live simulation components, provided by fielded command and control equipment, were able to interact with a constructive simulation called CIMUL8™ and a part task trainer (virtual simulation) for training in Multiple Launch Rocket System (MLRS) Fire Control Panel operations. Besides providing the first demonstration of its kind, this configuration was used for the purpose of evaluating a new training system (the MLRS Fire Control Panel Trainer (FCPT)) using real equipment inputs as well as inputs from a constructive simulation representation of the real equipment. The paper will describe the design of the evaluation, present some preliminary training evaluation results, and make recommendations for future use of the system for evaluation. The paper will also recommend additions to the DIS standards for better support of similar test systems.

About the Authors

Christina Bouwens is a project engineer with CAE-Link Corporation, Binghamton, New York. She has over five years experience in simulator networking and currently serves as systems engineer for several CAE-Link projects that use Distributed Interactive Simulation (DIS) and Aggregate Level Simulation Protocol (ALSP). Before joining CAE-Link, Ms. Bouwens was the program engineer for the DIS standards program at the Institute for Simulation and Training. Ms. Bouwens has a Masters of Science in Mathematical Science from the University of Central Florida and a Bachelor of Science in Mathematics from Geneva College. She has published several papers on the subject of DIS and interoperability. Ms. Bouwens chairs the DIS working group for Communication Architecture and Security.

Robert E. Jones, Jr. is a Senior Program Manager with CAE-Link Corporation. For the last 9 years, Mr. Jones has managed the Human Factors, MANPRINT, and training research projects for the Falls Church, VA office. These projects have been for a variety of clients including DoD, DoT, NRC and commercial industries. Previously, he served in the U.S. Army for 23 years as an Aviator and Research Coordinator. He has a BS in Mathematics and a M.Ed. in Educational Psychology.

Dr. Linda Pierce is the Ft. Sill Field Element Chief of the Army Research Laboratory, Human Research & Engineering Directorate. As a Research Psychologist, Dr. Pierce's background is in command and control research and selection and motivation in small groups. She is responsible for numerous projects for the Army Research Laboratory including development of a simulated environment in which human performance on currently fielded US Army command and control systems can be evaluated. In addition, she is developing a command and control staff effectiveness model using the JANUS simulation system. Dr. Pierce received her Ph.D. from Texas Tech University in Industrial and Organizational Psychology.

A DIS NETWORK FOR EVALUATING TRAINING SYSTEMS EFFECTIVENESS

Christina L. Bouwens and Robert E. Jones
CAE-Link Corporation
and
Dr. Linda Pierce
Army Research Laboratory Ft. Sill Field Element
Ft. Sill, Oklahoma

INTRODUCTION

In an era of rapid change and shrinking budgets, the use of simulations and simulators gives the military a capability to refine doctrine, test and validate tactics and modernize weapons systems to ensure things work *before* implementing changes. A variety of simulations and simulators are being developed by the Army in support of Field Artillery, fire support command and control, and weapon systems. Simulations and simulators are proving effective in decision making in system design; developing tactics, techniques, and procedures for system employment; identifying and resolving system-related MANPRINT issues; and providing more cost-effective training.

Many of the simulations and simulators developed to support either research or training were not designed to be interactive. The use of simulations or simulators in sterile, non-interactive environments increases the artificiality associated with the research and training and decreases the fidelity or external validity of both. Training individuals or individual crews to perform specific tasks does not guarantee they will have the ability to operate as members of a crew or task force. A Defense Science Board Report states:

> *"The Services train individual soldiers, sailors, airmen, and marines and provide highly trained combat units and do a very good job. [...But] some things we don't do well. First and foremost among these is the training and exercising of large, joint, or combined forces to fight on short notice."*[1]

What is required is a method to simulate the interconnectedness of battlefield operating systems -- command and control and weapons -- and the subsequent "fog of war" by introducing the same complexity and uncertainty into our use of simulations and simulators as will be apparent on the battlefield of the 21st century. The answer is in Advanced Distributed Simulation, as achieved through the Distributed Interactive Simulation standards:

> *"We believe that Advanced Distributed Simulation (ADS) technology is here today, and that this technology can provide the means to:*
>
> o *improve training and readiness substantially*
> o *create an environment for operational and technical innovation for revolutionary improvements*
> o *transform the acquisition process from within"*[2]

The U.S. Army Field Artillery School is currently involved in the development and fielding of fire support command and control systems and weapon systems. These systems are significantly better than predecessor systems, but are also significantly more complex to operate. At the same time, budgets are being slashed. Thus, it is imperative that more be done with less by developing simulations and simulators to be used during the acquisition and fielding of command and control systems such as the future fire control system, the Advanced Field Artillery Tactical Data System (AFATDS) and the Interim Fire Support Automation System (IFSAS), and current and future weapon systems such as the MLRS, Paladin, and the Advanced Field Artillery System (AFAS). This approach was supported by the Army Chief of Staff, General Gordon R. Sullivan, who in his presentation at the May 1993 AUSA Louisiana Maneuvers Symposium stated:

[1]Defense Science Board Report, "Impact of Advanced Distributed Simulation on Readiness, Training, and Prototyping," January 1993.

[2]Ibid.

"Distributed Interactive Simulations hold great promise for compressing the acquisition cycle and removing much of the frustration from our acquisition system. Simulation lets us see and touch the acquisition cycle. I believe we can collectively help change our heel-toe cold war system to a more responsive – and more cost effective – process."

To this end, the Army Research Laboratory (ARL) sponsored a research project to instrument the Depth and Simultaneous Attack (D & SA) Battle Lab with Fire Support command and control equipment. All equipment utilized in this project used the Distributed Interactive Simulation (DIS) protocols on a Local Area Network (LAN) and will eventually be connected to the Defense Simulation Internet (DSI). The purpose of this capability is to simulate realistic battlefield communications conditions for research and training. Devices that have been integrated into this simulation capability include: The CIMUL8™ / SPECT8™ / DISIP8™ (hereafter referred to as CIMUL8™) simulation system, two (2) Forward Entry Devices (FEDs), a Lightweight Computer Unit (LCU) running the MLRS Battery Fire Direction System (FDS) software, and a new desktop version of the MLRS Fire Control Panel Trainer (FCPT).

A second focus of this research project was to examine the extent to which training can benefit from this environment while retaining requirements for achieving established levels for proficiency. The integration of these devices onto an instrumented LAN permits the conduct of realistic Fire Support exercises which are able to be conducted in conjunction with any DIS compatible simulation using real soldiers performing tasks in the laboratory as they would in the field. Thus, the Army can begin to address, in a cost effective manner, issues related to doctrine, tactics, materiel, organization, leadership and soldiers before committing to doctrinal changes, costly acquisition programs, or extensive reorganizations. In particular, it permits evaluation of training devices in a simulated battlefield environment while allowing the collection of human performance data in a virtual setting.

SYSTEM REQUIREMENTS

The system developed for this program was required to link fielded command and control (C2) equipment, unmodified, to an MLRS FCPT device and a constructive simulation system called CIMUL8™. This integrated system allows the constructive simulation system to create and maintain a scenario consisting of simulated elements generated by the constructive simulation, the MLRS FCPT and the network interfaces to the C2 equipment (FEDS and the FDS). Operators of the C2 equipment transmit C2 data across the network to other C2 equipment and eventually to the MLRS FCPT. A data logger collects command and control information from the network for later analysis.

This network interconnection provided the MLRS FCPT trainer with realistic inputs to initiate an MLRS mission. The live C2 equipment is also available for training within the DIS environment.

Technical Requirements

The greatest technical challenge of this program was to provide a DIS interface for the command and control equipment. The interface had to be flexible and reusable for a number of devices. It also had to provide the ADS environment (created by the networking of simulations and equipment) the DIS information that is required by other participants but not normally part of the information generated by the C2 equipment.

Figure 1 shows the configuration of the network.

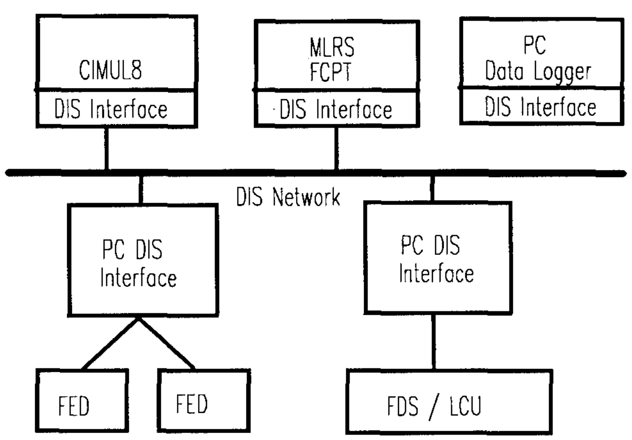


Figure 1
Network Configuration for ARL/Ft Sill

The PC interface had to provide a means to collect C2 signals from the C2 equipment, package it in a DIS Protocol Data Unit (PDU), and send it onto the local area network (LAN). It also had to do the reverse for incoming messages. DIS PDUs with command and control information were received by the interface from the network and the C2 text sent on to the C2 equipment.

In addition, the interface allowed for engagement of the entities operating the C2 equipment. That is, it interpreted incoming PDUs to determine if the entity had been affected by weapons effects. Although this does not enhance the MLRS FCPT training, it provides a less artificial representation of the C2 equipment.

Besides providing the interface for the C2 equipment, all devices on the network had to be integrated so that they would interoperate in a meaningful way. Implementation of the DIS PDUs alone does not ensure an interoperable system. Certain decisions had to be made as to how the PDUs would be implemented, what data should be included in the PDUs and the rules for when the PDUs would be issued. Integrated system testing would be required to ensure correct system operation using DIS.

Testing & Evaluation Requirements

The second focus of this research project was to examine the extent to which training can benefit from the DIS environment while retaining requirements for achieving established levels of proficiency. Outside of the DIS environment, inputs to the training system to be evaluated need to be generated from within the system. Evaluation is limited to what could be observed by the instructor or to that which is displayed on an instructor's station. For this project, a Training Effectiveness Evaluation (TEE) was conducted on the MLRS FCPT examining the feasibility and potential effectiveness of training soldiers of the future in the DIS environment. To perform the TEE, fire missions were transmitted over the network to an MLRS FCPT where an operator performed the missions. In this manner, appropriate targets were provided by the live equipment and engaged by the MLRS. The effects were evaluated using the displays of the constructive simulation, CIMUL8™.

In order to perform the TEE, there were a number of test and evaluation requirements:

Training Subjects

Students from the U.S. Army Field Artillery School were required to serve as subjects for the TEE.

Subject Matter Experts (SME)

SMEs were required to perform the following functions:

- o Review the experimental scenario for tactical accuracy and realism
- o Provide estimates of time expected for student completion of the training exercise
- o Evaluate the fidelity of the FCPT and the utility of the FCPT for training in a simulation environment

Data Collection Requirements

A number of datum needed to be collected to support the TEE. These included the following:

- o Response time data
- o Error Data (keystroke errors)
- o Student Questionnaires to determine student attitudes about training simulators
- o SME Questionnaires to determine estimates on expected student performance (time), the SME views on the fidelity and potential effectiveness of training on the FCPT in a simulation environment
- o Evaluation of the physical characteristics of the FCPT

INTERFACE/NETWORK DESIGN

Since this project was the first to provide a DIS interface for command and control equipment, there were no established guidelines to follow for developing the PC interface. Before this project, C2 equipment had been interfaced to PCs for the purpose of testing and stimulating the C2 equipment. This approach was utilized to provide an interface from the C2 equipment to the PC interface. The proper DIS PDUs were then created and sent out onto the LAN via the PC's Ethernet card.

The interface had to be able to accept information produced by the C2 equipment in its natural form. The C2 equipment utilizes a Frequency Shift Keying (FSK) modem. This modem connects directly to a radio through a special interface or to field wire using the two field wire connecting posts on the C2 device. We chose to use the field wire interface.

The PC interface needed to have an FSK modem that could connect to the field wire interface. Since FSK modems are not commercial off-the-shelf (COTS) items a special modem had to be utilized. Such FSK modem boards had been available in the past for PCs running SCO UNIX. Since our system was running under DOS, we required the same functionality for a DOS environment. TELOS was developing a DOS version of the modem board and provided a Beta version of the TELOS Signal Master™ board as the FSK modem for the PCs. Two boards were used, each with two communication channels. One board served the two FEDs. The other board was dedicated to the FDS which required two communications channels; the 6-character net which provided communications to the FEDs and the 11-character net which provided communications to the launcher. This approach is shown in Figure 2.

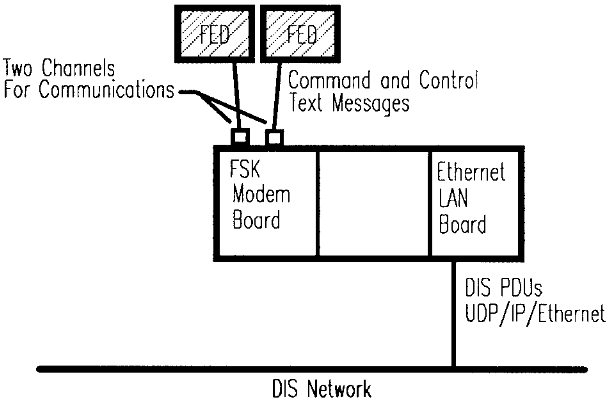


Figure 2
PC Interface to C2 Equipment

In this configuration, the FEDs would send a C2 message through the two wire interface to the FSK modem board of the PC interface. The modem board "strips off" the communication protocols and stores the ASCII text of the C2 message in memory. The developed interface software reads the ASCII text message and sends out the appropriate transmitter and signal PDUs to the DIS network via the Ethernet LAN board. The ASCII text message is included as data in the signal PDU.

Two DIS standards were implemented. The DIS PDU standard version 2.0.3 was used for the PDU formats. The draft standard for Communication Architecture for DIS was also used. The widespread use of both standards provided assurance of maximum compatibility with other systems. An Interface Control Document (ICD) was developed for the project to specify the DIS interface requirements for all the systems on the network. This ensured that the individual systems implemented the DIS standards in a consistent manner. These requirements are summarized below:

The PDU standard was implemented with the following guidelines and assumptions:

- o Only the DIS PDUs required to simulate the integrated system were implemented. These included: Entity State, Fire, Detonation, Transmitter, and Signal PDUs.
- o Command and Control messages were communicated using Transmitter and Signal PDUs. Receiver PDUs were not required for this implementation.
- o Entity State PDUs were issued on behalf of the entity containing or controlling the transmitting device (FED or FDS), for the MLRS launcher, and additional friendly and opposing forces represented by the constructive simulation. Articulated parts were not represented.
- o Fire and Detonation information associated with the munition fired by the MLRS FCPT was communicated using the Fire PDU and Detonation PDU. In addition, positional information and movement of the launcher represented by the FCPT were represented using Entity State PDUs.
- o Simulation of the actual radios along with associated jamming, noise, interference, etc. was not represented in this integrated system. It is assumed that the devices send and receive perfect signals. The interface will distinguish between radio frequencies, therefore incoming PDUs must show the correct frequency in order to be passed on to the C2 equipment.
- o FED and FDS related entities did not maneuver while the simulation was running. There is an offline capability to "Beam" the FED and FDS entities to various locations on the battlefield.

The Communication Architecture for DIS (CADIS) draft standard version 1.0 was chosen for use with this interface. Protocols used on the local area network were:

Application, Presentation & Session Layers:	DIS 2.0.3
Transport & Network Layer:	UDP/IP (CADIS 1.0)
Data Link Layer:	Ethernet
Physical Layer:	Ethernet

After the individual systems had implemented the DIS standard according to the system ICD, a week of integration testing was performed to ensure that CIMUL8™, the MLRS FCPT, and the C2 equipment were able to interoperate correctly.

A technical demonstration was carried out to show the interoperability of the system. A scenario was developed demonstrating close operations beginning with target identification by a Forward Observer (FO) and ending with the launch of six rockets by the MLRS FCPT. One FED was used to represent the FO and the other FED represented the Fire Support Team (FST). The MLRS Fire Battery Fire Direction Center was represented with the Lightweight Computer Unit (LCU) running the Battery FDS software. MLRS launcher actions were simulated by the MLRS FCPT. CIMUL8 provided a graphics view of the battle, using icons to shown the location of the various participants (FO, FIST, FDC and launcher) along with simulation of other friendly and opposing forces. In addition, weapons fire was graphically displayed by CIMUL8 based on the receipt of DIS PDUs from the network.

The sequence of events for this demonstration is shown in Figure 3.

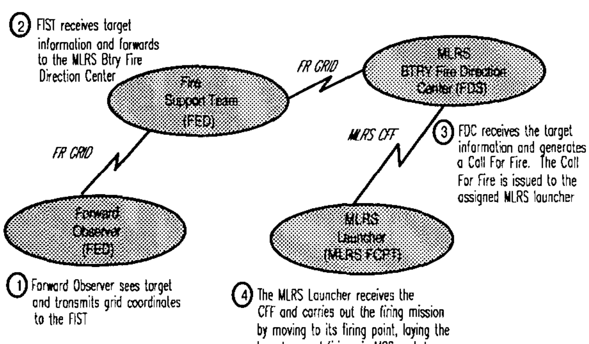


Figure 3
Scenario Events for System Demo

TRAINING EFFECTIVENESS EVALUATION DESIGN

Since this was the first known use of DIS for training system evaluation, there were no designs to follow. Standard TEE type data were gathered by adapting a network data logger to gather training data from the network instead of directly from the training device. This affords the advantage that the training device does not have to be especially equipped or programmed for data collection if it already has a DIS network interface. It also allows the collection of such data from the live equipment on the network, potentially allowing an investigation of training effectiveness of the actual equipment.

A total of 30 soldiers, half of whom were MLRS Fire Direction Center (FDC) operators and the other half MLRS crew members, served to support the TEE as student subjects. Each of the students trained on the FCPT in the configuration described in previous paragraphs. Data on the soldier performance was gathered through a combination of collected network data and evaluator observation.

SMEs from the Gunnery Department, most of whom were MLRS instructors, also supported the TEE. The SMEs reviewed the experimental scenario for tactical accuracy and realism. They also provided time criterion estimates used to evaluate soldiers' performance, evaluated the fidelity of the FCPT, and furnished input about the utility of the FCPT for training in the simulation environment.

Data collected for the TEE included the following:

Time Data: Response time data was automatically captured by a PC datalogging program as each student proceeded through three experimental simulation scenario runs on the FCPT. The data logger time-tagged the point in time at which the subject at the FCPT transmitted the Fire Mission "Will Comply" signal back to the Battery FDC indicating fire mission start time. The data logger also time-tagged the point at which the soldier fired the first rocket of the mission, indicating fire mission end time. There were two fire missions per simulation run. Response time data were defined as the amount of time soldiers required to successfully perform all steps of a fire mission. The criterion times were subjectively established by the SMEs, based on their experience.

Error Data: Error data was obtained through observation. As each soldier proceeded through the experimental simulation scenarios on the FCPT, a trained research assistant noted any keystroke errors committed by the soldier, and recorded these on a standard data collection form. It would be advantageous for future experiments to have the error data collected by the data logger as well.

Student Questionnaire. Questionnaires were administered to soldiers at the end of each training session. The questionnaire assessed soldiers' attitude about training simulators and their views on the FCPT in the DIS environment.

SME Questionnaire. SMEs typically reported in groups of two or three to evaluate the FCPT in the simulation environment. They were briefed on the simulation system and the purpose of their participation in the study. Following this introduction, each SME was given the opportunity to proceed through the same simulation scenario on the FCPT that soldiers experienced as well as performing any other actions that they wanted to perform on the FCPT. After completing their exercises, SMEs were asked to provide estimates of the expected performance time for soldiers performing the experimental simulation scenarios so that soldiers' performance time data could be evaluated relative to a performance standard.

Evaluation of the Physical Characteristics of the FCPT. A human factors evaluation of the physical characteristics of the FCPT was also conducted. The critical internal components of the FCPT including the disk drive and internal computer components, the fidelity of the FCPT screen, and the soldier-machine interface were examined by a Human Factors Specialist.

THE EXPERIMENT

Although the FEDs were a part of the network, in order to reduce the number of operators required to carry out the experiment, CIMUL8™ was used to generate the FR GRID (Fire Request using Grid Coordinates) messages that would normally be sent by the FEDs. The experimental simulation scenario proceeded as follows:

- CIMUL8™ initiated a force-on-force battle simulation. A few minutes into the battle, CIMUL8™ generated command and control FR GRID messages that were transmitted onto the network and received as CALL FOR FIRE (CFF) messages at the Battery Fire Direction Center (FDC). The first CFF was then relayed to the FCPT as a Fire Mission.

- The receipt of the fire mission by the FCPT was indicated by an alarm signal, which meant the MLRS FCPT had received a C2 message. The soldier was required to respond by pressing the appropriate keys that would cause a "WILL COMPLY" message to be issued to the FDS. At this point in the scenario, the Self-Propelled Launcher Loader (SPLL) was positioned at a Hide Point. The soldier was required to perform all the necessary keystrokes to move the SPLL to the Fire Point requested by the FDC and then fire the mission.

- After firing the mission and performing the proper keystrokes to stow the weapon, the solder performed the keystrokes necessary to move the SPLL to a second Hide Point as requested by the FDC, at which point the soldier received a second Fire Mission. The soldier then performed all the keystrokes necessary to move the SPLL to a second Fire Point as requested, and fire a second mission. After stowing the weapon, the mission was ended and the first run concluded.

- Each soldier repeated the experimental simulation scenario three times.

EXPERIMENTAL CONCLUSIONS

Results

A variety of analyses were performed on the data collected including: a) analyses of variance on the performance time data collected by the data logger, and error data collected through observation, b) descriptive statistics for the time, error, and questionnaire data, and c) content analyses for the open-ended questionnaire items.

The analyses of the time data showed that there was a clear learning trend (Figure 4). Soldiers required substantially less time to perform the fire missions with increased practice over the three scenario runs. It is also noteworthy that a much greater percentage of soldiers were able to meet the performance time criterion in Run 3 (87%) compared to Run 2 (70%) and Run 1 (30%).

By the final scenario run, soldiers performed their missions almost flawlessly (Figure 5). A much greater percentage of soldiers committed no errors by Run 3 (64%) as compared with Run 2 (32%) and Run 1 (4.5%). This is particularly meaningful since the error rate decreases concurrently with decreases in response time. Thus, no time-error tradeoff was demonstrated.

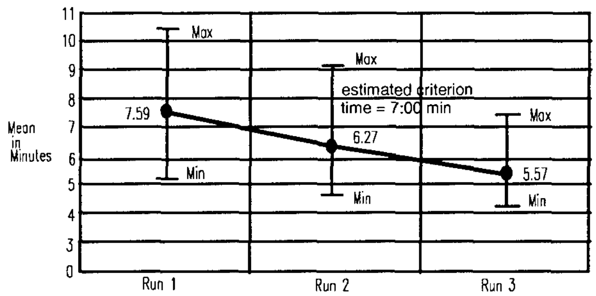


Figure 4
Response times for students

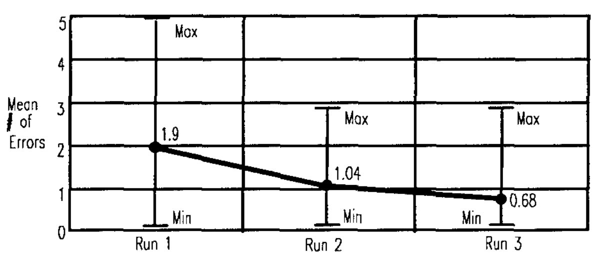


Figure 5
Errors Committed Students

In response to the questionnaires, soldiers viewed their training on the FCPT in a simulation network environment very positively and would recommend this type of training for fellow soldiers. This was not surprising based on the expressed views of solders during their training sessions. On average, they seemed both curious and excited about the prospects of future training in such an environment.

The SMEs also viewed the FCPT in the simulation network environment very favorably. Their responses indicated a positive attitude about the utility of the FCPT as a training device that, in a simulation environment, could effectively supplement and maintain soldier training. The SMEs also indicated that this environment provided added realism with FDS and launcher interactions represented. They felt that because tasks such as communications with the FDC and fire missions were more lifelike, soldiers would develop a greater level of confidence in their abilities to perform these tasks in the field.

The SMEs also pointed out that training on the FCPT in this environment could supplement field training and classroom knowledge and could be used without extensive planning (e.g. training on an "as needed" basis). They also saw the system providing a potentially significant training benefit to the National Guard units who, because of future time and training cost constraints, may not otherwise have sufficient hands-on training opportunities.

As part of the TEE, a human factors evaluation was conducted on the physical characteristics of the FCPT. The device presented a realistic view of the actual FCP except for simulation of vehicle moves. The FCPT currently simulates vehicle moves by trainees pressing a "MOVE" button on the panel. This method, although better than an "automatic move", in which vehicle moves are not simulated at all, does not provide users with a visual image of the move action as it occurs. It is therefore recommended that the FCPT be upgraded to include a graphic display of vehicle move actions that would create a higher degree of realism.

Based on the information gathered during the TEE, the feasibility and effectiveness of training in the DIS environment appears promising. Data collected on soldier performance clearly demonstrate that significant early learning occurs for students training on the MLRS FCPT in the network environment.

The general outlook that soldiers and SMEs had toward the training system was also encouraging. Soldiers viewed training on the FCPT in this DIS environment very positively and felt confident that it could help them and others do their job more effectively. SMEs believe that the integrated FCPT provides superior realism and could serve as an effective training tool for supplementing early learning as well as refresher training.

Future Work

Aside from the abundance of positive information that was gathered over the course of the TEE, some potential research initiatives clearly remain. Most notable during the TEE was the lack of an automated data collection capability. The data collection process was substantially limited by the current data collection capabilities of the DIS implementation. The data that were captured represent a small fraction of data available for capture. The potential certainly exists (e.g., Kaye & Copenhaver (1992)) for automated collection, reduction and analysis of a variety of performance data including total time, mission segment time, keystrokes, errors (including when they occur), and accuracy. The proposed system should also have the flexibility to allow the insertion of system-specific performance measures (i.e., specific to the device on the DIS network) that could supply feedback to the student and allow instructors to track student performance. This data collection system would provide a means to obtain and analyze performance data from the operation of any military device that is integrated into the DIS network.

The Simulation Management (SIMAN) PDU development in the DIS community has taken a significant step toward addressing the data capture requirements for experimental use of a DIS network. The next step is to build this capability into the trainer itself in order to send key datum for collection via the DIS network.

Further exploration is also required in the area of integrating other real world command and control devices into the DIS network. By design, the interface used to support the current C2 devices can be implemented to support other C2 devices, and could serve as a means to test the interaction of new and developing command, control and communications (C3) technologies. The interface could also be utilized to allow new and developing systems (e.g. Advanced Field Artillery Tactical Data System (AFATDS), the Improved Data Modem (IDM), and the Aviation Mission Planning System (AMPS)) to test their capabilities with existing systems in the DIS environment.

DIS Accomplishments and Recommendations

The DIS network interface developed for the FEDs and the FDS represents the first time that real, unmodified battlefield command and control equipment has been interfaced to the synthetic environment using DIS. This has allowed real equipment to operate with simulations in a laboratory environment. With the eventual installation of a Wide Area Network (WAN) the lab assets will have the capability to participate in DIS exercises with participants located at remote locations.

This effort served as a proof of principle that: a) a training device can be successfully integrated into a DIS environment together with actual military command and control devices and b) performance data can be captured and analyzed from a training device operating in that environment.

In conclusion, a significant step has been taken toward bringing real world command and control systems into the synthetic environment for training, testing, evaluation, and data collection purposes. This TEE has provided a unique opportunity to investigate another advantage of DIS applications. The present findings provide the basis for further exploring the DIS environment as a training and research instrument.

We believe that DIS technology holds great promise for the future of training and system evaluation.

ACKNOWLEDGMENTS

Work discussed in this paper was performed under contract to the U. S. Army Research Laboratory, Contract #MDA903-92-D-0039, D.O. #008.

CIMUL8™, SPECT8™, and DISIP8™ are trademarks of BDM Federal Systems.

Signal Master™ is a trademark of Telos Corporation.

REFERENCES

Copenhaver, M. and Ching, H. "Training Effectiveness Evaluation of an MLRS Fire Control Panel Trainer Using Distributed Interactive Simulation," U.S. Army Research Laboratory Draft Technical Report, Feb. 1994.

Bouwens, C. and Ching, H. "Development and Engineering of a Distributed Interactive Simulation System," U.S. Army Research Laboratory Draft Technical Report, Feb. 1994.

DIS Steering Committee, "The DIS Vision: A Map to the Future of Distributed Simulation," Comment Draft, Oct. 1993.

"Proposed IEEE Standard Draft: Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications, Version 2.0 Third Draft", IST-CR-93-15, May 1993.

"Final Draft Proposed IEEE Standard: Communication Architecture for Distributed Interactive Simulation" IST-CR-93-20, June 1993.

Application of GPS to Hybrid Live/Constructive/Virtual Training Systems

R. J. Van Wechel
R. P. Jarrell
Interstate Electronics Corp.
Anaheim, California

ABSTRACT

GPS user equipment has matured and is now available to support the use of live players in networked live/constructive/virtual wargaming simulations. GPS provides true WGS-84 based coordinate information anywhere in the world at any time and to accuracies at the 5 ft (1σ) level (demonstrated in high dynamic aircraft using differential GPS).

In supporting DIS-based hybrid live/constructive/virtual networked team training, GPS is directly applicable to the dead reckoning requirements of DIS. The on-board state vector for an integrated GPS/Inertial Reference Unit provides accurate position, velocity and acceleration as well as attitude and attitude rate information so that dead reckoning thresholds can be both position and attitude driven. A simplified analysis is presented in the paper to derive dead reckoning update rates from the G loading levels of various player dynamics. Also, information is provided which results in word length requirements for GPS-based state vector information for transmission over minimum word length DIS Field Instrumentation Protocol Data Units (PDUs, which are the data block formats). The coordinate frame problem in use of GPS-based state vector information from fixed ranges is also addressed, showing that the use of a local geodetic frame is preferable to the use of an earth centered earth fixed frame, in that it is more efficient of network PDU word length. Weapon scoring requirements using GPS-based state vectors are addressed in terms of GPS state vector accuracy required to score various weapons and provide "positive training". These requirements are all applicable to the JTCTS and NGTCS programs which are in the formative stages and will use GPS-based information in DIS Field Instrumentation PDUs.

Results are presented of a combined Northrop/IEC demonstration using the China Lake RAJPO GPS assets linked into a DIS demo for I/ITSEC in November 1993.

ABOUT THE AUTHORS

Robert J. Van Wechel is Senior Chief Scientist at Interstate Electronics Corporation (IEC). He is one of the originators of the IEC all-digital GPS receiver architecture, having been involved in GPS activities at IEC since the mid-1970s. He served on the IEC team in support of the TRW team for the TCTS phase I effort, and has also done system analysis studies for JACTS employing GPS and means of inserting DIS techniques into air combat training systems.

Richard Patrick (Pat) Jarrell is the Section Manager for Navigation Systems Analysis and the TCTS Program Manager at Interstate Electronics Corporation. Mr Jarrell came to IEC in 1988 after retiring from the US Navy. During his Naval career he served as a Strategic and Tactical Weapons Officer in the submarine community and completed his career by serving as the Commanding Officer of the FBM Navigation Test Unit on board the USNS Vanguard. Mr Jarrell has worked in the areas of GPS and Acoustic navigation systems accuracy analysis since working for IEC.

Application of GPS to Hybrid Live/Constructive/Virtual Training Systems

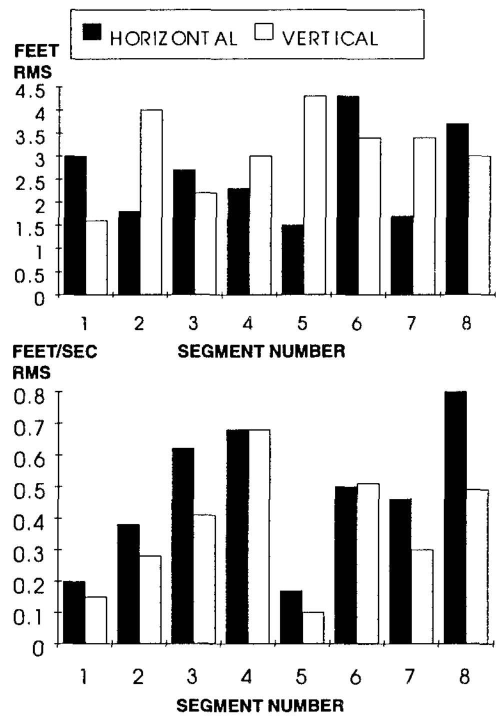
R. J. Van Wechel
R. P. Jarrell
Interstate Electronics Corp.
Anaheim, California

INTRODUCTION

The Global Positioning System (GPS), which is now operational, consists of a network of 24 satellites which provide true WGS-84 based three-dimensional position and time on a continuous basis with world-wide coverage[1,2,3]. Although the overall and usually quoted GPS position accuracy is 16 meters (52.5 feet) SEP (Spherical Error Probable), experience in most applications has been considerably better than this. During the Desert Storm operation from 15 January to 3 March 1991, long term averages over 11,000 navigation solutions showed the average SEP to be 8.3 meters[3] (27.2 feet), rather than the 16 meter specification. With differential GPS[4], which uses a ground reference receiver to remove common mode errors between the reference station and the user (such as satellite orbit and clock errors, and the common mode portion of ionospheric errors), errors of 2 meters or less are routinely achieved.

GPS therefore provides an invaluable tool in instrumenting live platforms on training ranges. In addition to the high accuracies, operation is achieved on high dynamic platforms at G loads to 8 Gs. For high dynamic platforms, the GPS receiver is best integrated with an inertial reference unit. GPS and inertial systems are highly synergistic; the GPS removes the troublesome biases of low cost/low quality inertial systems, and the inertial system carries the GPS receiver through high dynamic maneuvers and temporary signal blockages caused by shadowing of the antennas.

Typical accuracies of differential GPS in a high dynamic platform using method one[4], which is by far the best because it accounts for rapid satellite switching in high dynamic maneuvers, are shown in figure 1, which shows horizontal and vertical position and velocity errors in both F-15 and F-16 flight tests[5]. These particular flight tests were conducted by the Tri-Service GPS Range Applications Program Program Office (RAJPO) on the High Dynamic Instrumentation Set (HDIS) developed for that program by Interstate Electronics Corporation.



SEGMENT KEY

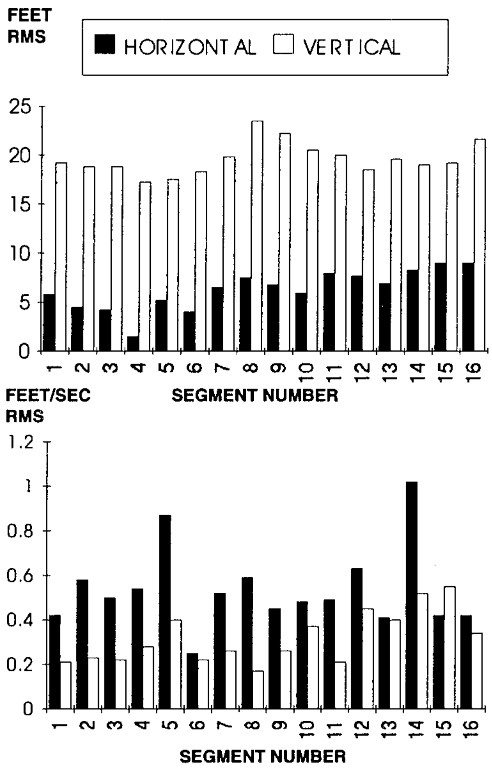
1. STRAIGHT & LEVEL	5. STRAIGHT & LEVEL
2. FIGURE EIGHT CW	6. CLIMB & DIVE
3. TWO CIRCLES CW	7. FOUR POINT ROLLS
4. CUBAN EIGHT	8. RACE TRACK

ERROR SPECIFICATIONS

HORIZ POS - 6 FEET RMS	HORIZ VEL - 1.6 F/S RMS
VERT POS - 12 FEET RMS	VERT VEL - 2.7 F/S RMS

Figure 1. Flight Test Results[5] for differential GPS with inertial aiding

Absolute accuracy test results are shown in figure 2. The dynamics in these tests covered the full range up to 8 Gs in the various maneuvers used with intermittent satellite visibility caused by antenna shadowing. Ground truth was provided to 2 ft. accuracy at Eglin Air Force Base by a truth system consisting of cinetheodolites, laser ranging, FPS-16 radars and aircraft inertial navigation systems.



SEGMENT KEY

1. STRAIGHT & LEVEL	9. STRAIGHT & LEVEL
2. FIGURE EIGHT CW	10. STRAIGHT & LEVEL
3. TWO CIRCLES CW	11. FIGURE EIGHT
4. STRAIGHT & LEVEL	12. TWO CIRCLES CW
5. RACETRACK CCW	13. STRAIGHT & LEVEL
6. STRAIGHT & LEVEL	14. RACETRACK CCW
7. FIGURE EIGHT CW	15. STRAIGHT & LEVEL
8. TWO CIRCLES CCW	16. STRAIGHT & LEVEL

ERROR SPECIFICATIONS

HORIZ POS - 21 FEET	HORIZ VEL - 1.6 F/S
VERT POS - 35 FEET	VERT VEL - 2.7 F/S

Figure 2. Flight Test Results[5] for absolute GPS with inertial aiding

APPLICATION TO TEAM TRAINING

The way in which GPS will be used in large-scale training systems is currently being defined by key programs such as JTCTS and MAIS. An effort to develop standards for these applications is being carried on by the Field Instrumentation Working Group which is a part of the Distributed Interactive Simulation (DIS) standards development activity. Currently, a draft standard for field instrumentation use has been prepared and is in review. The key driver in developing these standards is minimizing the data that needs to be sent on the data link, since the data link has proven to be the main bottleneck in instrumenting large training exercises.

DIS Dead Reckoning Algorithms

A key part of the DIS standards is a technique which is very useful with regard to efficiently loading a data link with time-space-position information (TSPI). This algorithm is shown in the block diagram of figure 3. The GPS receiver output, integrated with an inertial reference unit so it periodically measures both position and attitude, is compared with the output of a "dead reckoning model". This model is a time extrapolation of previous outputs of the GPS receiver ($x = x_o + v_o t + a_o t^2$).

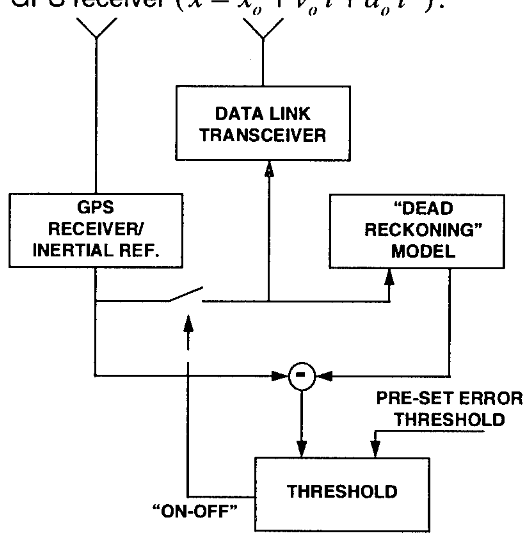


Figure 3. Dead Reckoning Algorithm for efficiently loading a data link with TSPI

When the error between the GPS-measured position (and attitude) and the extrapolated position (and attitude) exceeds some threshold level (like 10 or 20 ft.), the GPS measured state is input to both the data link and the dead reckoning model. At the distant data link receiver terminal, a similar dead reckoning model is also updated with a new state vector. The two dead reckoning model errors are thereby corrected and they can then run for a while at acceptable error levels until again updated. The algorithm thus provides a means of minimizing data link loading, according to the activity level of the aircraft. When the aircraft is flying with minimal acceleration, the extrapolation can run much longer than when it is going through high-dynamic maneuvers. It therefore provides a means of taking advantage of the fact that most of the aircraft in an exercise are not maneuvering, and can be sampled at a low rate, and those that are in high dynamic maneuvers are automatically sampled at a high rate.

This algorithm is ideal for training systems because it allows supporting very high fidelity weapon simulations by closing down the error threshold for certain exercises requiring high precision such as no-drop-bomb-scoring (NDBS). It is even possible to close some portions of the error down more than others, such as closing down the vertical error more than others for NDBS, since NDBS is particularly sensitive to vertical error. The possibility also exists of closing the error threshold down only when required, such as during weapon launch, and when the aircraft is paired as a target.

One of the key questions with regard to the use of this algorithm in a large scale training system is how fast it will update vs. platform dynamics and threshold level settings. The answer to this question heavily influences data link requirements, and also affects instrumentation accuracy levels. In order to provide an easily analyzable case to demonstrate results and to provide some upper bounds on update time or "rules of thumb" as a guide for system design, the aircraft (or other platform) can be assumed to be flying in circular turns at a constant G loading. Sections of circular turns are common in aerobatic maneuvers. Even though a complete circle is not flown, small sections of circles are representative of sections of aerobatic maneuvers, since aircraft are constrained by the laws of physics to fly in a circular path constrained by G loading. More complex maneuvers can be considered as being made up of sections of circular turns. Other platforms, such as ships, also commonly make circular turns.

By using this simple maneuver as a basis for analysis, it is possible to easily derive the upper bounds for update times for various dead reckoning models versus their G loading and threshold values. In the case of aircraft (and ships), the roll angle is ignored in this analysis, and only pitch and yaw angles are considered. Also, angle-of-attack and angle-of-sideslip are assumed constant, such as would occur in a steady-state circular turn. Although these assumptions may appear to limit the validity of the analysis, they allow arriving at first-order approximations which have been shown to be consistent with flight simulator test results, although somewhat optimistic, since they are more in the nature of bounds.

Using this approach, these update times were calculated as outlined in appendix A and are presented in figure 4 for various G loadings, position thresholds and attitude thresholds.

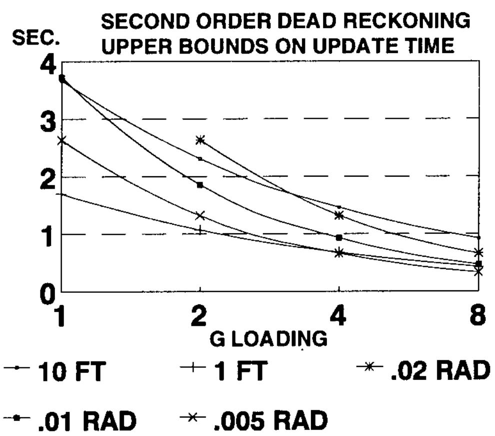


Figure 4. TSPI Update Time Bounds for Position and Attitude Threshold Second Order Dead Reckoning

An interesting comparison of these results can be made with data from the Northrop Corporation Flight Simulation Laboratory[6]. Using thresholds of 9.1 feet in position, 3 degrees in attitude, and second order dead reckoning, they obtained an average of about 1.2 updates per second for second order dead reckoning in high dynamic air combat scenarios, and an average of 2.42 updates per second for the same scenarios with first order dead reckoning.

It would be beneficial to the field instrumentation community if further similar studies were done by organizations doing dead reckoning in large scale simulations. This would allow further refinement of these bounding update time levels.

TSPI Word Length Requirements

Considerable effort has been expended by the Field Instrumentation Working Group of the DIS in the past two years in attempting to reduce the PDU (Protocol Data Unit) sizes for field instrumentation. The standard DIS PDUs are much too large to be feasible for range data link use. For example, the standard DIS entity state PDU, which defines the state vector of a platform, contains 64 bit position and velocity words. In this regard, the requirements of integrated GPS/Inertial systems for word length should be taken into account, because these are the TSPI sensors that will be used for the indefinite future in these systems. As shown in the above test data, instrumentation accuracies of close to 1 foot in position and close to 0.1 ft/sec in velocity are achievable. There is then no point in much more word length in the PDUs than that which will support 1 foot instrumentation. Attitude accuracies of 0.1 to 0.2 degrees are achieved by these same instrumentation systems, which should set the resolution of the attitude information in the DIS PDUs. Assuming maximum values such as would be required for a training system such as TCTS, the word lengths shown in table 1 result from these instrumentation accuracy levels. It should be noted that these word lengths are much less than those in the current DIS standard.

The Field Instrumentation Working Group of the DIS has proposed a flexible format of "Profiles" for the field instrumentation PDUs[7] which would have the flexibility to support variations in applications dictated by various ranges. This would be implemented by table-driven software where the profiles for each application provide the control for parsing the PDU data. With this approach, the use of word lengths no longer than the instrumentation will support should be possible.

Table 1. TSPI Word Length Requirements for a typical range training application

Full Scale Values	Resolution of LSB	Word Length Required
X,Y - 2761 nmi. =16777216 ft.	1 foot	24 bits
Height - 65536 ft.	1 foot	16 bits
Velocity - +/-3277 ft/sec	0.1 ft/sec	16 bits
Acceleration - +/- 10 Gs	300 uG ~ .01 ft/sec2	16 bits
Attitude - 360 degrees	0.05 degree	13 bits

Coordinate Frames for TSPI

The choice of coordinate frame for use in the field instrumentation PDUs also affects the number of bits required in the PDUs. For a purely surface-based exercise, there is no need to send height information, since the position coordinates provide all of the required information. Even for aircraft, fewer bits are required for instrumenting height above a range than the horizontal dimensions of the range, as seen in table 1. In comparing the use of ECEF coordinates with geodetic coordinates, studies have shown[8] that for a hypothetical 155 x 155 NM x 65000 ft range considered, 55 bits are required (for 4 ft resolution) for the three coordinates of position for translated ECEF coordinates, regardless of whether 2-dimensional or 3-dimensional information is required. For translated geodetic coordinates, 51 bits are required for the three position coordinates if height information is required (3-dimensional case), and only 37 bits are required if height information is not required (2-dimensional case). In the 2-dimensional case, velocity and acceleration state bit requirements are also significantly reduced when geodetic coordinates are used, since no height terms are necessary. This is not to say that ECEF or some other coordinate frame is not used internally for computation. ECEF coordinates are simply not the most efficient frame to use for data transmission.

Weapon Scoring for positive training

In applying GPS to training such as an ACMI system, where the positions of the shooter and target, and the attitude of the shooter must be instrumented to provide information to weapon simulations, it is necessary to define TSPI accuracies (this determines whether absolute or differential GPS is required) for the GPS instrumentation system which will result in positive training. There is no clear definition of what constitutes positive training. For example, if guns are to be used which require 0.1 degree attitude accuracy in the shooting platform, positive training for gun scoring cannot result if only a 2 degree instrumentation system is used. There might as well not be any instrumentation because the scoring can be predicted as well by a coin toss.

There appear to be no guidelines currently to define what TSPI accuracies are required. A suggested approach for unguided ballistic weapons is contained in Appendix B. It is hoped that the training community will soon address this problem, and in addition the corresponding case for guided weapons.

Demonstrations

In November 1993, a demonstration of the use of GPS was conducted by the Northrop Corporation and IEC, using the RAJPO GPS assets at the China Lake Naval Weapons site. Data from the China Lake site was sent by phone line to the Northrop facility at Hawthorne, California, where dead reckoning was applied to the data. Also additional simulated aircraft were flown with the live aircraft in the same scenario. This was an initial demonstration of the use of live and constructive platforms together. It was demonstrated at the 1993 I/ITSEC.

CONCLUSIONS

The GPS satellites are now in place, the user equipment is available, and the DIS is proceeding to define the Field Instrumentation PDUs. The pieces are coming together for the next generation of live/constructive/virtual training systems.

REFERENCES

1. L. Graviss, *GPS Development Program Status*, Proceedings of the Institute of Navigation GPS-92 Fifth International Technical Meeting of the Satellite Division, Sept. 16-18, 1992, pp. 3-16.

2. Freer, *GPS Operational Status*, Proceedings of the Institute of Navigation GPS-92 Fifth International Technical Meeting of the Satellite Division, Sept. 16-18, 1992, pp. 17-22.

3. Sharrett, Wysocki, Freeland, Brown, and Netherland, *GPS Performance: An Initial Assessment*, NAVIGATION: Journal of the Institute of Navigation, Vol. 39, No. 1, Spring 1992, pp. 1-24.

4. E. Blackwell, *Overview of Differential GPS Methods*, NAVIGATION: Journal of The Institute of Navigation, Vol. 32, No. 2, Summer 1985, pp. 114-125.

5. S. Mahmood and C. Simpson, *Integrated DGPS/Inertial TSPI for DOD Ranges: RAJPO Equipment Test Results*, Proceedings of the 1993 National Technical Meeting, Institute of Navigation, Jan. 20-22, 1993.

6. Goel and Morris, *Dead Reckoning for Aircraft in Distributed Interactive Simulation*, Northrop Corporation, 1992, and American Institute of Aeronautics and Astronautics.

7. *Standard for Information Technology, Protocols for Distributed Interactive Simulation Applications*, Version 2.0.3 FI, Draft Standard, 3 February, 1994.

8. Boehme and Van Wechel, *Coordinate Frames for Field Instrumentation PDUs*, Presented to the Ninth Workshop on Standards for the Interoperability of Defense Simulations, Sept 13-17, 1993.

APPENDIX A
TSPI UPDATE TIME BOUNDS FOR DEAD RECKONING

To derive the update times for the dead reckoning algorithm, the aircraft is assumed to be flying at a constant airspeed, and to be making turns having various G loadings. In these circular turns, assume the position of the aircraft to be described by the vector

$$\bar{r} = R(\cos\,\omega t\,\bar{i} + \sin\,\omega t\,\bar{j}) = R\bar{\rho}$$

where $\bar{r} =$ aircraft position vector
$\bar{\rho} =$ unit radius vector
$\bar{i}\,, \bar{j} =$ unit x and y vectors
$R =$ turning radius
$\omega =$ rate of turn (radians/sec)

The velocity vector is the derivative of this, or

$$\bar{\upsilon} = \omega R(-\sin\,\omega t\,\bar{i} + \cos\,\omega t\,\bar{j}) = V\bar{\nu}$$

where $\bar{\nu} =$ unit velocity vector
$V = \omega R$

The acceleration is, by differentiating again,

$$\bar{\alpha} = -\omega^2 R(\cos\,\omega t\,\bar{i} + \sin\,\omega t\,\bar{j})$$
$$= -\omega^2 R\bar{\rho} = -A\bar{\rho}$$

where $A = \omega^2 R = V^2 / R$

The jerk or next derivative is needed, because it is the lowest order rate which is not measured, and therefore contributes the most error. It is

$$\bar{\varsigma} = \omega^3 R(\sin\,\omega t\,\bar{i} - \cos\,\omega t\,\bar{j})$$
$$= \frac{-A^2\bar{\nu}}{V} = -J\bar{\nu}$$

The position error due to the jerk will then be

$$\varepsilon = \frac{J}{6}\Delta t^3 = \frac{A^2}{6V}\Delta t^3$$

We can then calculate the update time that the TSPI algorithm will operate at as a result of the position error to be approximately (setting the threshold equal to the position error)

$$\Delta t = \sqrt[3]{\frac{6\varepsilon V}{A^2}}$$

The algorithm of figure 3 also can operate on an attitude error threshold. In the DIS dead reckoning approach, a combined position and attitude error threshold is used. If either position or attitude error exceeds thresholds, the dead reckoning model is updated.

The attitude of the aircraft is constantly changing in the turn and is represented as the unit velocity vector

$$\bar{\nu} = -\sin\,\omega t\,\bar{i} + \cos\,\omega t\,\bar{j}$$

The attitude rate is the derivative of this, or

$$\bar{\psi} = -\omega(\cos\,\omega t\,\bar{i} + \sin\,\omega t\,\bar{j})$$

The second derivative of attitude is then

$$\bar{\mu} = \omega^2(\sin\,\omega t\,\bar{i} - \cos\,\omega t\,\bar{j}) = -\frac{A^2}{V^2}\bar{\nu}$$

If the attitude threshold is then θ_t,

$$\theta_t = \frac{1}{2}\frac{A^2}{V^2}\Delta t^2$$

the update time that the TSPI algorithm will operate at as a result of the attitude error will be approximately (setting the threshold equal to the attitude error)

$$\Delta t = \sqrt{\frac{2\theta_t V^2}{A^2}}$$

The update time for the combined position and attitude error threshold criteria will then be the smaller of the two or

$$\Delta t = Min\left[\sqrt[3]{\frac{6\varepsilon V}{A^2}}, \sqrt{\frac{2\theta_t V^2}{A^2}}\right]$$

These update times are calculated and plotted in figure 4 for various G loadings, position thresholds and attitude thresholds.

This approach allows a quick comparison to be made of first and second order dead reckoning. If only position and velocity are used (first order), the acceleration is not instrumented. The position error due to the acceleration not being instrumented is then

$$\varepsilon = \frac{A}{2} \Delta t^2$$

The update time that the TSPI algorithm will operate at as a result of the position error is then approximately (setting the threshold equal to the position error)

$$\Delta t = \sqrt{\frac{2\varepsilon}{A}}$$

To compare this first order dead reckoning model with the second order model, take the case of a 5 ft. threshold at 8 Gs. For the first order case shown here, the result is 0.198 seconds, as compared to 0.728 seconds for the second order dead reckoning model. This demonstrates the value of the second order model in reducing the sample rate required.

The use of circular turns at constant G loading provides a simple and convenient analysis tool for dead reckoning models. Update time bounds for specific G loads can be analytically determined, and can be verified by simulation or flight test. With this, the improvement of the second order dead reckoning model can be demonstrated analytically.

APPENDIX B
TSPI ACCURACY REQUIREMENTS FOR UNGUIDED WEAPON SCORING

Introduction

The purpose of this appendix is to outline a means of evaluating instrumentation accuracy requirements for unguided ballistic weapon scoring in simulated weapon delivery training exercises. In evaluating scoring effectiveness, the effectiveness of the following are considered:

- The effectiveness of the particular weapon against the target of interest, given perfect performance by the operator of the weapon system.

- The effectiveness of an instrumentation system in scoring simulated weapon firings of the weapon against the target of interest.

- The scoring of the trainee in firing the weapon against the target of interest.

The first parameter of importance relates to target size and other characteristics, as they relate to the weapon used against it. This includes target overall dimensions as well as particular areas of the target that are most vulnerable to the weapon. For example, this implies the dimensions of the engine room area of a ship, and its vulnerability to a bomb or torpedo.

In this discussion, we roll all of these target size /weapon effectiveness characteristics into effectively an increased target size with a dimension we call t, as shown in figure B-1. This target size parameter must be determined from the combination of target and weapon characteristics as outlined above, and may also be dependent upon target orientation.

The second parameter deals with weapon dispersion effects that are uncontrollable by the trainee. These effects should be distinguished from the dispersion effects that the training program is attempting to reduce. In dropping dumb bombs, this parameter should include unknown dispersing effects, or effects that the trainee is not expected to account for, such as wind changes (if he is accounting for wind in some way, such as in high altitude bombing), local gravity anomalies (again this probably only applies to very high altitude bombing and to ballistic missiles). When an automatic bomb release system is used, the errors of the automatic system should be included here, since they represent errors that would occur even if the trainee performed his procedure perfectly.

Uncontrolled dispersion effects are then rolled up in this parameter and called σ_d, with dimensions the same as target size t. This is illustrated in the diagram of figure B-1.

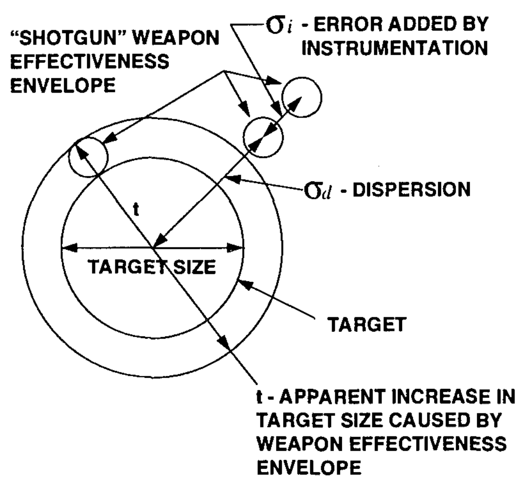


Figure B-1. Target Size and Error Definitions

Next are the instrumentation errors in instrumenting the trainee. These are termed σ_i. This term includes the uncertainty in both target and shooter position and attitude (and how their derivatives integrate into position and attitude error), since relative position and attitude are the parameters of interest.

The last parameter is the dispersion measure of the trainee. With a gun, this is his aiming error when firing, and is the principle measure of his training program with that weapon. Since instrumentation error is the main interest here, the error of the trainee is not discussed further. That is the subject of the training program which uses the instrumentation system.

Given these parameters, an analysis can be performed to determine various probabilities. The probabilities of interest are:

- The probability of a hit (P_h) given perfect performance by the trainee. This is dependent upon the first two measures t and σ_d. The probability of a hit is the probability that the weapon will hit inside the target dimension area including the effective size increase caused by the weapon effectiveness envelope (the trainee doesn't need to be so accurate if the weapon effectiveness envelope is large). It can be argued that a training program gives realistic, positive training if the probability of a hit and miss match the probability obtained if the trainee performs perfectly. These first probabilities are a measure of effectiveness of the weapon against the target of interest.

- The probability of a hit (P_{hi}) when the instrumentation system is added (includes σ_i as well as t and σ_d) is the next probability of interest. The comparison of this probability with the above probability is a measure of effectiveness of the <u>training instrumentation system</u>. <u>If the instrumentation system significantly changes the probabilities determined by t and σ_d alone, then the training system does not properly score the trainee, and the system probably provides negative training.</u>

- The final probabilities are the probabilities of a hit and miss when σ_a, the aiming error of the trainee is included. This is the score of the trainee. The best he can do is to match the probabilities computed from t and σ_d alone, that is, he matches the probabilities of a hit and miss (and the kill probability) of the weapon against the target of interest. This probability is not analyzed here, but could be done in a training program.

The next step in this process is to determine the probabilities from the σ measures. Then specific numbers must be applied for each weapon and target combination of interest. Note that it is not necessary to perform this analysis on every target/weapon combination, but to perform it on the most stringent cases that the training instrumentation system must handle. This should only require analysis of a few cases.

Analysis
We initially wish to determine the effectiveness of the particular weapon against the target of interest, given perfect performance by the operator of the weapon system. The parameters outlined above to define this are the target size t, and the dispersion σ_d. We assume the dispersion to be normally distributed. The probability of a hit is then defined by the normal probability integral

$$P_h = \frac{1}{\sqrt{2\pi}} \int_{-\frac{t}{2\sigma_d}}^{\frac{t}{2\sigma_d}} e^{-x^2/2} dx$$

This integral is the area under the normal probability curve (figure B-2) between the limits + and - $t/2\sigma_d$. The density function has been normalized to unit variance.

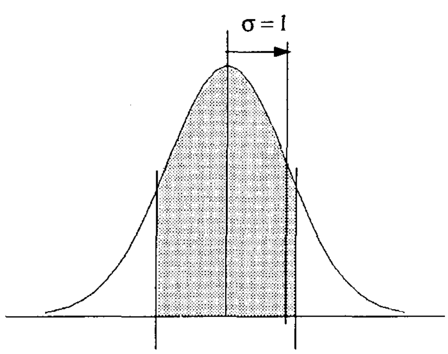


Figure B-2. Normal Probability Integral

When the instrumentation error σ_i is added, the dispersion σ_d increases to

$$\sigma_{di} = \sqrt{\sigma_d^2 + \sigma_i^2}$$

With this increase in overall dispersion due to instrumentation, the hit probability decreases to a value P_{hi}. These probabilities have been calculated as a function of dispersion and instrumentation error, all normalized to target size. As an example, if the instrumentation error is 0.553 of the target size, $\sigma_i / t = 0.553$. Next, supposing that the ratio of dispersion to target size is 0.4, $\sigma_d / t = 0.4$. The combination of dispersion and instrumentation error then yields $(\sigma_d^2 + \sigma_i^2)^{1/2} / t = 0.6825$. The term $t / (2(\sigma_d^2 + \sigma_i^2)^{1/2})$, which is the upper and lower integration limit of the Gaussian distribution is then equal to 0.733. Without any instrumentation, the upper and lower integration limit is $t / 2\sigma_d = 1.25$ (calculated from $\sigma_d / t = 0.4$). From tables of the normal probability integral, the probability of a hit without the instrumentation is then $P_h = 0.789$ and the probability of a hit with the instrumentation is $P_{hi} = 0.536$. The probability of a hit is then reduced by a factor of $P_{hi}/P_h = 0.536/0.789 = 0.68$ by the instrumentation. We define negative training to occur when this probability ratio goes below some pre-determined threshold value.

Using this approach, the instrumentation error normalized to target size (σ_i/t) is plotted versus the dispersion normalized to target size (σ_d/t) in figure B-3, for the defined values of the probability ratio P_{hi}/P_h. At low values of σ_d/t, for which $P_h = 1$, which represents the case of weapons which have high accuracy relative to target size, the probability ratio P_{hi}/P_h is the same as would be predicted for hit probability given only the instrumentation error and target size. Thus, for $\sigma_i/t = 0.5$, the hit probability is 0.68. For inaccurate weapons, represented by large values of σ_d/t, the 1-sigma instrumentation error can grow to larger values.

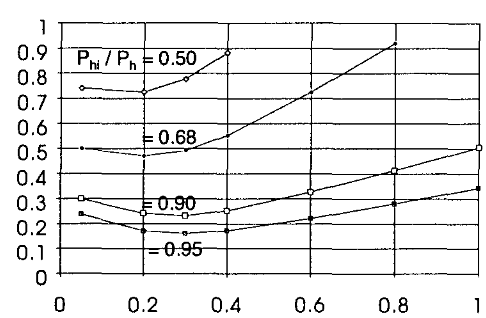


Figure B-3. Instrumentation Error vs. P_{hi}/P_h

This plot also shows that the approach of making instrumentation error equal to dispersion is consistent with this approach, providing dispersion is not less than about 1/3 the target size, and P_{hi}/P_h is between 0.68 and 0.9. Making instrumentation error equal to dispersion is equivalent to a value of P_{hi}/P_h equal to about 0.7 or 0.75.

Figure B-3 allows some simple "rules-of-thumb" to be derived for the relationship of instrumentation error to target size and weapon dispersion. For low dispersion weapons, and a high probability of positive training given by $P_{hi}/P_h = 0.95$, the one-sigma instrumentation error must be about 1/5 of the target size. It can grow to about 1/3 the target size when the dispersion is equal to target size. Following these guidelines assures that the instrumentation system will degrade the hit probability to only 95%, giving a high degree of positive training.

For $P_{hi}/P_h = 0.68$, and low dispersion weapons, the one-sigma instrumentation error can be made equal to half the target size. This is a compromise solution, providing less than ideal training in scoring, but defines the degradation in training to a prescribed level.

A few specific cases are discussed below. A value of $P_{hi}/P_h = 0.68$ is used.

Air-to-surface guns - For a surface target such as a tank or armored personnel carrier, the target size is about 12 by 26 ft (size of a typical tank) . Guns typically have a "cone of fire" which is designed into the gun to ease the aiming problem. This "cone" is distinguished from dispersion because it increases the probability of a hit on the target when successive rounds are fired. A typical angular width for this cone is 4 milliradians. At a range of 1500 ft, this cone subtends about 6 ft. It thereby increases the effective target size somewhat to approximately 20 by 35 ft (adding about 3/4 of this cone size on each side of the target). We therefore approximate the target size as about 20 by 35 ft.

The actual dispersion, as defined, is very low, so the 1-sigma instrumentation error must be about 1/2 the target size, or from 10 to 18 ft (1-sigma) at 1500 ft range. Attitude error in instrumenting the shooter must be 10 to 18/1500 or 6.7 to 12 milliradians 1-sigma (0.4 to .7 degrees 1-sigma). These allowable errors are actually on the high side of what would be ideal, since the probability of a hit was reduced to 68%, and the target size has been "stretched" to the limit. Using $P_{hi}/P_h = 0.95$ would require much better instrumentation, since then the one-sigma instrumentation error would need to be about 1/5 of the target size or 3.5 to 7 ft (1-sigma).

Air-to-air guns - For air-to-air guns, a very similar situation exists. Target size for a tail view of a fighter (a typical shooter/target pairing orientation) is about 9 ft by 43 ft (aircraft height by wingspan). Assuming the same 4 mil firing cone, at 1200 ft range, the firing cone subtends about 5 ft. The effective target size might be increased by the firing cone to about 16 by 50 ft. This yields a positional 1-sigma error of 8 to 25 ft (for the half-target size criteria). Attitude accuracy required also follows the preceding example, requiring about 8 to 25/1200 or 6.7 x 21 mils or .4 x 1.2 degrees 1-sigma error. Again, these errors are on the high side of what would ideally be allowed.

Conclusions

The basic premise of this analysis is that the instrumentation system should give essentially the same probability of a hit as the real weapon would against the real target. We have allowed the hit probability to deteriorate to 0.68 (or to any desired fraction) of what it would in the real situation (0.68 is a compromise - ideally 0.90 or 0.95 would be used). This is the definition of positive training in weapon scoring proposed here. If the probability of a hit deteriorates to much worse than this as compared to the real situation, the training is more of a random numbers game, and it is doubtful that the trainee will benefit much from the experience, at least in weapon proficiency.

SIMULATION MANAGEMENT IN

DISTRIBUTED INTERACTIVE SIMULATION

Huat K. Ng, Ronald S. Klasky, Kenneth P. Kelly
Veda Incorporated
Orlando, Florida

ABSTRACT

The standardization of simulation protocols through the SIMulator NETworking (SIMNET) and Distributed Interactive Simulation (DIS) concepts has allowed the interconnection of dissimilar simulators into an electronic battlefield. A distributed simulation may encompass many different types of systems and the number of entities in an exercise can grow to thousands. As the network traffic increases, an exercise control and management system is critical in order to successfully control the scenario. In DIS, a host computer designated as the Simulation Manager (SM) performs exercise management functions via 12 SImulation MANagement (SIMAN) PDUs. Some functions performed by the SM include: Start, Restart, Maintenance, and Shutdown of an exercise. The focus of this paper is to describe an on-going design and development effort which will result in the test, validation and implementation of the 12 new SIMAN PDUs on a workstation. From this workstation, a DIS exercise manager will be able to utilize the SM software to control all of the entities (live, constructive and virtual) on the battlefield.

ABOUT THE AUTHORS

Huat K. Ng received his Bachelor's degree in Computer Engineering and Master's degree in Electrical Engineering from the University of Central Florida. Mr. Ng has worked with Veda Incorporated since 1993 on realtime simulation networking. He has been involved with Distributed Interactive Simulation (DIS) standard development and implementation of application level gateways to interface dissimilar simulation protocols.

Ronald S. Klasky is a visual systems engineer with Veda Incorporated. Mr. Klasky has a Master's degree in Computer Science, specializing in computer graphics and seven years of graphics programming experience. His duties include developing computer graphics software in support of simulation projects in such areas as terrain database display and graphical user interfaces.

Kenneth P. Kelly has over 13 years in support of various NAWC-TSD and STRICOM programs. He is the Project Manager on Veda's Reconfigurable Ground Vehicle Test Bed project and the DIS Simulation Management effort. Mr. Kelly has a Bachelor's degree in Industrial Engineering and a Master's degree in Engineering Management. He also serves as the Orlando Branch Manager for Veda Incorporated.

SIMULATION MANAGEMENT IN

DISTRIBUTED INTERACTIVE SIMULATION

Huat K. Ng, Ronald S. Klasky, Kenneth P. Kelly
Veda Incorporated
Orlando, Florida

INTRODUCTION

The standardization of simulation protocols via the Distributed Interactive Simulation (DIS) protocol has demonstrated the feasibility of interconnecting multiple distributed simulators via a local/wide area network. The Simulation, Training and Instrumentation Command's (STRICOM) Battlefield Distributed Simulation - Developmental (BDS-D) program has begun to address the extension of creating an entire electronic battlefield. The BDS-D goal is to develop the ability to network geographically separated simulation sites while preserving time/space coherence and synchronization within the limits of human perception and reaction times. In order to successfully manage the entire electronic battlefield of the BDS-D program, an exercise control management capability must be added to the electronic network.

A distributed simulation may encompass many different types of simulator systems and the number of entities in an exercise can grow to the hundreds or thousands of entities. As the number of entities expands, an exercise control and management system is critical in order to successfully control the scenario. In DIS, a host computer designated as the Simulation Manager (SM) performs exercise management functions. Some functions performed by the SM include: start, restart, maintenance, and shutdown of exercises. Other functions provided by the SM include the introduction of late players and the collection and distribution of specific types of data. In the most recent unapproved DIS standard (version 2.0.3) [1], 17 new Protocol Data Units (PDUs) were proposed. Among the 17 new PDUs, 12 of the PDUs will be used for simulation management activities. An initial DIS architecture has been presented in the DIS Communication Architecture Subgroup (CAS) and the simulation management activities serve to establish a portion of the Session Database. The Session Database is defined in the DIS architecture as "a standard database which includes network initialization data and simulation entity initialization and control data" [1]. SIMAN PDUs have not yet reached maturity. However, the PDUs have been defined in a manner to provide flexibility and extensibility.

Multiple SMs may exist in a distributed simulation exercise, with an individual SM controlling an individual site or simulation. The multiple SMs may be arranged in a hierarchical structure with each having different responsibilities and levels of authority. The DIS standard does not bar the use of multiple SM hosts to perform exercise control and management. An example of a hierarchical structure of multiple SMs is depicted in Figure 1.0.

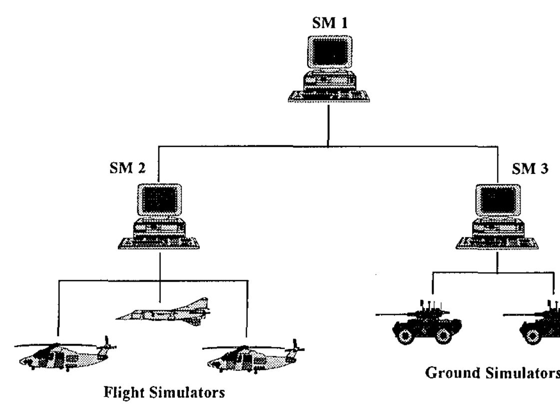


Figure 1.0 Example of a Hierarchical Structure of Multiple Simulation Managers

BACKGROUND

The DIS standard has been approved by the Institute of Electrical and Electronics Engineering (IEEE) as the IEEE P1278 standard which addresses the communication protocols employed between simulation applications. The current state of the simulation entity is conveyed to other simulators using a PDU. A simulation entity is defined as "an element of the synthetic environment that is created and controlled by a simulation application through the exchange of DIS PDUs" [1]. At present, the IEEE P1278 defines only ten PDUs. These PDUs are related to entity interaction and logistics information.

As the DIS standard evolves, new PDUs are added as required. In the most recent unapproved DIS standard, i.e. Version 2.0 Third Draft (or 2.0.3), 17 new PDUs were added to the original IEEE P1278 standard. The DIS PDUs defined in Version 2.0.3 can be logically organized into six different protocol families:

- entity information/interaction
- warfare
- logistics
- simulation management
- distributed emission regeneration
- radio communications

Among the 17 new PDUs described, 12 of the PDUs will be used for simulation management activities. The DIS 2.0.3 will be submitted to the IEEE in mid 1994. The goal of the SIMAN PDUs is to provide a centralized control mechanism of the simulation exercise.

Veda Incorporated, under STRICOM sponsorship, has developed a software package called DISMAN (Distributed Interactive Simulation MANager) to provide distributed simulation management capabilities. DISMAN adheres to the DIS standard 2.0.3. Due to the infancy of the SIMAN PDUs, the enumerated list for datum identifications is short (less than 50) and incomplete in DIS 2.0.3. The enumerated list consists of numerical values associated with each simulation attribute. The enumerated list, however, is being expanded and finalized in the DIS working groups, which consist of industry, academia, and military representatives. The standardization process is continuing with DIS standard workshops held biannually under the sponsorship of STRICOM.

The research, design and implementation of a stand-alone exercise control station, utilizing the 12 SIMAN PDUs, is the focus of this project. DISMAN will be used in the Anti-Armor Advanced Tactical Demonstration (A^2ATD) Number One. Experiments are also being discussed for the 1994 Interservice/Industry Training Systems and Education Conference (I/ITSEC) to provide exercise control, initialization of entities, and automated data collection. The design goals of DISMAN is to provide the flexibility to add future requirements such as the Session Manager. The Session Manager is a planned future expansion of the SM.

SIMULATION MANAGEMENT PDUs

The SIMAN PDUs are described in document [1] to support managerial chores in a simulation exercise. DISMAN will utilize the 12 SIMAN PDUs and will serve to establish a portion of the Session Database for simulations participating in a DIS exercise. The Session Database defines its contents in two major categories: Network data and Simulation Entity data. Within the Session Database, DISMAN falls into the category of Simulation Entity data. The Simulation Entity data category includes the information needed to initialize all of the simulation entities with the required parameters and data to support a specific exercise.

The 12 SIMAN PDUs are as follows:

- Create Entity
- Remove Entity
- Start/Resume
- Stop/Freeze
- Acknowledge
- Action Request
- Action Response
- Data Query
- Set Data
- Data
- Event
- Message

Functionalities of DISMAN

The primary functionality of DISMAN is to use the SIMAN PDUs to actively manage simulation hosts and applications. Management of a simulation includes:

1. maintaining and updating the local simulation session databases on simulation hosts,
2. configuring the simulation applications and entities it simulates,
3. performing diagnostics and other maintenance procedures on simulation hosts and applications,
4. coordinating the entity's involvement in an exercise, and
5. monitoring the applications and entities for signs of problems which the manager should address.

Using the SIMAN PDUs, the manager can set, control, initiate an action, automate data collection, and report an event on an entity or a group of entities. Examples of the usage of the SIMAN PDUs follow:

- Parameters that can be set by DISMAN include location, velocity, orientation, dead-reckoning algorithm, force identification, and entity type. These parameters are defined in the Set Data PDU.
- Controlling an entity's state is accomplished by using the Stop/Freeze, and Start/Resume PDUs.
- An action such as turning on or off VV&A data collection by a simulation asset is accomplished via the Action Request PDU.
- Automating data collection for after action review and analysis is accomplished with the Data Query PDU. The Data Query PDU provides the flexibility in setting the frequency of a periodic data response by the simulator. DISMAN can also be used to set event conditions, so that information that is of interest will be sent from the simulator onto the network when a condition is flagged true.

DIS is an application layer protocol, and to verify reliable protocol handshaking there is a corresponding pair of PDUs for each simulation management transaction. For example, the Acknowledge PDU will be used by DISMAN to acknowledge the simulator's receipt of the Start/Resume and Stop/Freeze PDU. Figure 2.0 illustrates the transaction involving the Stop/Freeze and Acknowledge PDU.

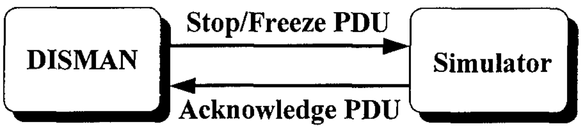


Figure 2.0 Example of a Simulation Management PDU Transaction

Simulation management functions in the DIS protocol are not strictly realtime. Functions such as initialization and configuration management transpire before an entity joins a realtime DIS simulation exercise. To begin an exercise, DISMAN will transmit a Start/Resume PDU with the real-world and simulation time embedded into the PDU. The real-world time corresponds to the time (with respect to Greenwich time) at which the entity is to start/resume the exercise. The simulation time corresponds to the time of day in the simulated world at which the entity will start/resume the exercise. In management functions that involves monitoring and exercise coordination, DISMAN allows the user to specify a lead-time in the corresponding SIMAN PDU. For example, if the manager of an exercise would like to monitor the fuel quantity of a simulator, a Data Query PDU will be sent by DISMAN to the simulator. The Data Query PDU will include a set time interval which specifies the time interval between issues of Data PDUs by the simulator.

Simulation management functions typically do not generate a vast amount of data traffic. Unlike the Entity State PDUs, the SIMAN PDUs do not follow a dead-reckoning algorithm to determine the frequency of PDU transmission. Simulation management functions are transaction oriented. A request from DISMAN invokes a single response from each managed entity. The only exception is in the case of a periodic data response to a Data Query PDU. However, even then, the frequency of transmission is still not as high when compared to a moving entity's amount of Entity State PDU traffic. Due to the non-realtime, low traffic characteristics of the simulation management protocols, DISMAN does not need to be defined in a manner that minimizes PDU sizes or processing requirements. Therefore, DISMAN is designed with the goal of providing the manager maximum flexibility and extensibility.

Data Requirement Set

The 12 SIMAN PDUs are defined in the proposed DIS Standard 2.0.3. Each SIMAN PDU is preceded with a simulation management header which consists of a PDU header (similar to other PDU families), the originating entity, and the intended receiving entity. The overall length of a simulation management PDU header is 28 bytes.

In order to convey the information between DISMAN and simulation applications, data are represented as identity/value pairs. Each identity/value pair is encapsulated in a fixed datum or variable datum record. To provide the flexibility in communicating information, SIMAN PDUs (Action Request, Action Response, Data Query, Data, Set Data, Event Report, and Message) are defined in a manner that allows concatenation of a variable number of datum records.

The identity of the individual datum fields are defined in [2]. This field is defined as a 32-bit enumeration. For example, the ammunition quantity identity is currently defined in DIS 2.0.3 to be enumeration value 40. Upon receiving a datum identity of value 40, the simulator recognizes that the correponding datum value is the ammunition quantity. The enumerated list is currently minimally defined, ie. only 50 items. However, as the standard evolves, enumeration values can easily be added without redefining the overall PDU structure.

DISMAN SOFTWARE DESIGN

Several factors were taken into consideration in the software design for DISMAN. The software must provide an interface that is easy and intuitive to the manager, yet still provide all the functionalities required by the defined simulation management protocols. DISMAN must be capable of supporting large simulation exercises and must convey important managerial information to the DIS manager. Although the simulation management protocols are not strictly realtime, the manager needs to obtain information regarding the simulation entities in a quick and efficient manner.

To accommodate the list of enumerated datum identities in the DIS 2.0.3 simulation management protocols, DISMAN provides a Graphical User Interface (GUI) to the user. The simulation management parameters are configurable and are entered into DISMAN in a format consistent with the SIMAN PDUs. With the aid of a pull-down menu system the user can easily navigate and select individual datums without the need to refer to the entire defined enumerated list. This feature hides the details of the PDU construction, but still provides the user with a powerful tool.

A Plan-View Display (PVD) is provided to the manager. The PVD displays a terrain database with UTM grid lines. Some of the map features that can be selected for display are trees, roads, lakes, rivers, and contour lines. The coordinate system can be selected to be map-defined UTM or user-defined pseudo-UTM. Simulation entities on the network appear on the PVD as icons with or without their entity identification (site id, application id, entity id) as labels. The mouse is used to select individual entities for the purpose of conveying information via PDUs. Entities can be individually selected or grouped by a "click-and-drag" feature of the mouse. Figure 3.0 illustrates the PVD.

Overall Architecture

The utility of a layered model in describing the overall architecture of DISMAN promotes an architecture that clearly defines each layer's definition and functionalities. In the field of network communications, layering concepts such as the International Standards Organization (ISO) Open Systems Interconnection (OSI) model provides a solution for interoperability of different systems.

The overall DISMAN architecture is composed of three layers: application, assembly, and network. The application layer implements the simulation management functions. The application layer also interprets and performs any acknowledgements to SIMAN PDUs received from the network. A GUI/PVD is provided to the user for ease of accomplishing the SIMAN applications. The

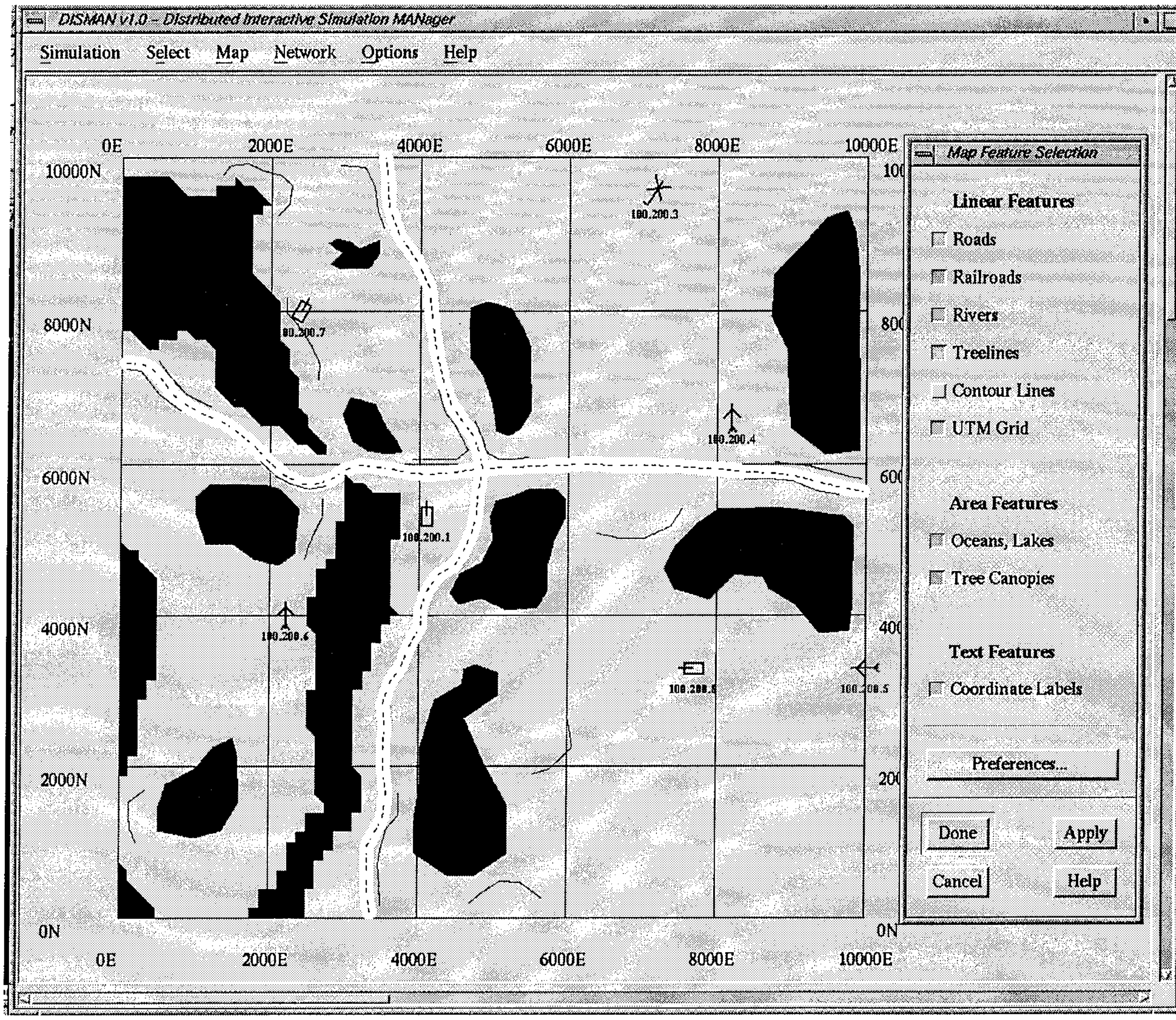


Figure 3.0 DISMAN's Plan-View Display

assembly layer performs the construction and decomposition of the SIMAN PDU. Data inputs from the application layer will be assembled into a SIMAN PDU structure as defined in the DIS 2.0.3 standard. Similarly, the assembly layer will disassemble a DIS PDU (SIMAN and Entity State PDUs) from the network into an internal data structure for further processing by the application layer. The assembly layer provides the interface between the application and network layer via service access points. Finally, the network layer performs the functions necessary to support the communication protocol stack. The network layer provides the necessary communication protocols such as UDP/IP/Ethernet. The Internet protocol stack of UDP/IP/Ethernet follows the recommended Phase 0 DIS communication architecture [3].

The advantage of utilizing a layered approach is to provide a clear definition in functionalities and to encourage software modularity. The value added to by modularizing the software is the ability to break up the software design into manageable pieces that are loosely coupled with each other.

Software Approach

To provide maximum code reuse, the design of DISMAN employs modern software practices. Issues such as code modularity, strong prototyping, and code readability are major criteria in the software approach. With the tremendous advances in today's computing hardware, care is being taken in the software approach to promote code portability and reusability between computer systems. Software designs that effectively alienate the software solution from the hardware to the maximum extent possible can reduce the cost of maintaining the software across different computing platforms. Software that is designed to be extensible, and capable in absorbing changes to requirements such as the changes resulting from the evolution of the DIS standard is highly desirable.

DISMAN is written in ANSI-C on a Silicon Graphics Indy workstation. The software approach employs standard Unix networking libraries. The socket system call is used to obtain a socket descriptor for performing networking tasks. Standard Interprocess Communication (IPC) libraries are utilized to provide communications between processes. DISMAN is divided into four concurrent processes:

1) Low-Level I/O
2) Protocol Translator
3) Entity Update
4) Graphical User Interface

Figure 4.0 illustrates the four processes that reside in the DISMAN software.

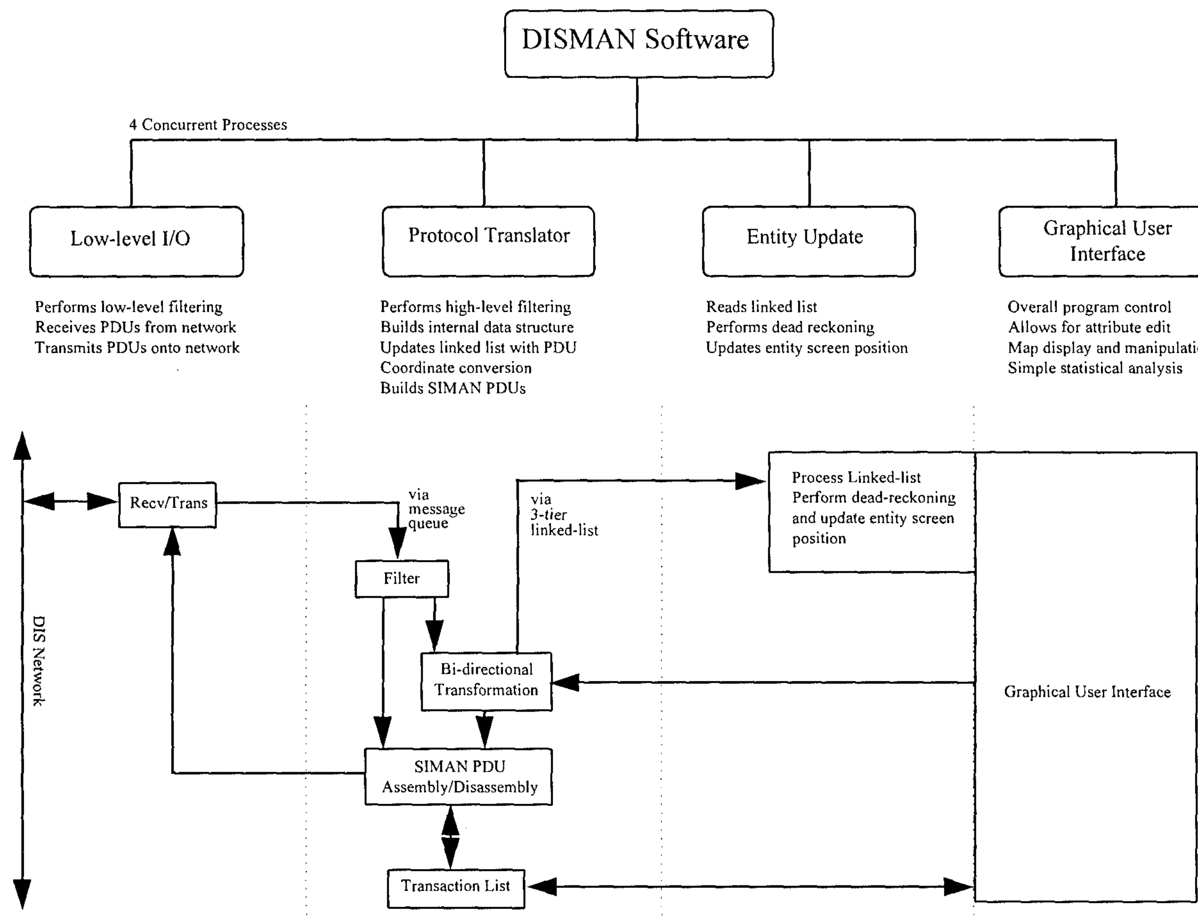


Figure 4.0 DISMAN Software Architecture

The Low-Level I/O performs filtering of PDUs based on the port identification number (DIS uses UDP port number 6993). Any data received other than DIS data will be discarded without further burden to the CPU. On the output side, when DISMAN transmits a SIMAN PDU, the PDU will be encapsulated by a UDP/IP/Ethernet frame before transmission onto the network.

The Protocol Translator process performs several functions such as high-level filtering, bidirectional transformations, updating the DISMAN internal link list, and builds the appropriate SIMAN PDUs. The most CPU intensive operation in the Protocol Translator process is the bidirectional transformation between DIS PDUs and internal DISMAN data structure format. Once the data are translated, they are entered into a three-tier linked list data structure for the Entity Update process. Similarly, in translating from a DISMAN data structure into a SIMAN PDU, the reverse process in coordinate conversion is applied.

The Entity Update process traverses the three-tier linked list data structure and performs dead reckoning and entity screen position updates on each item. The advantage of using a three-tier linked list is to provide another level of filtering based on entity identification (site, application, entity) values. DISMAN can selectively group the entities based on any combination of site, application and entity identities. Figure 5.0 illustrates the three-tier linked list data structure utilized in DISMAN.

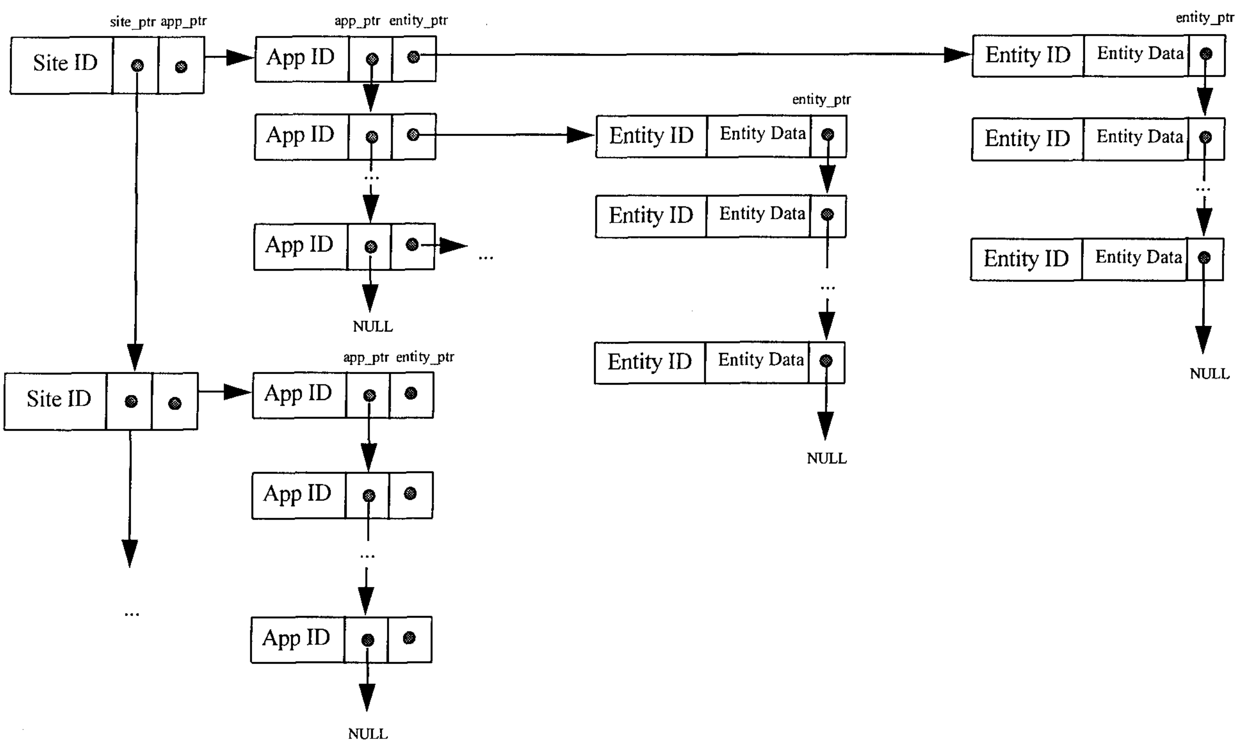


Figure 5.0 The Three-Tier Linked List Data Structure in DISMAN

The Graphical User Interface (GUI) process provides the user with a front-end system to perform overall program control, attribute edits, map display, and statistical analysis. The GUI is written in C utilizing standard X/Motif graphical libraries. Map control features such as contour lines, tree lines, tree canopies, rivers, UTM grid lines, panning, zooming are provided to the user. Pull down and pop-up menus are provided along with the map to enter entity attributes. The GUI provides the user an intuitive way in which to run DISMAN without the need to read software manuals or to understand the SIMAN PDU formats.

CONCLUSION

The DISMAN software utilizes standard software libraries to promote code reusability across different computational platforms. Modular software coding practices are employed in the software approach to allow ease in changes and add-on features. With the DIS standard still evolving, changes to DISMAN enumerated datum list will occur in the near future. DISMAN can be easily modified to meet these changes due to the modularity of the design. By providing the user with a simple-to-use GUI feature, DISMAN hides the details of the SIMAN PDUs, yet equips the manager with a powerful management tool.

The DISMAN program has provided the DIS community with a simulation management tool. By automating the managerial tasks by utilizing the 12 SIMAN PDUs, DISMAN has provided the manager with a systematic approach to simulation management. DISMAN will be used in the first A^2ATD demonstration to control exercise scenario. As of the date of this paper, testing of the DISMAN is still in process and the upper limits (number of entities) that DISMAN can control have not been determined. A^2ATD will be the first experiment which incorporates DISMAN as the simulation manager/controller. In A^2ATD, DISMAN will be used to control manned simulators, such as the M1A2. The next series of planned experiments will test DISMAN with Semi-Automated Forces (SAF) systems such as STRICOM's ModSAF (Modular Semi-Automated Forces) and CGF (Computer Generated Forces). These tests will show the utility and necessity of simulation management tools in controlling manned and automated forces.

LIST OF ACRONYMS

A^2ATD	Anti-Armor Advanced Tactical Demonstration
BDS-D	Battlefield Distributed Simulation - Developmental
CAS	Communication Architecture Subgroup
CGF	Computer Generated Forces
DARPA	Defense Advanced Research Projects Agency
DIS	Distributed Interactive Simulation
DISMAN	Distributed Interactive Simulation MANager
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Standards Organization
ModSAF	Modular Semi-Automated Forces
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PVD	Plan-View Display
SAF	Semi-Automated Forces
SIMAN	SImulation MANagement
SIMNET	SIMulation NETwork
SM	Simulation Manager
STRICOM	Simulation, Training and Instrumentation Command
UTM	Universal Transverse Mercator

BIBLIOGRAPHY

[1] "Proposed IEEE Standard Draft Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications, Version 2.0, Third Draft", *Technical Report IST-CR-93-15,* Institute for Simulation and Training, University of Central Florida, May 1993.

[2] "Enumeration and Bit Encoded Values for Use with Protocols for Distributed Interactive Simulation Applications", *Technical Report IST-CR-93-19,* Institute for Simulation and Training, University of Central Florida, June 1993.

[3] "Communication Architecture for Distributed Interactive Simulation (CADIS)", *Technical Report IST-CR-93-20,* Institute for Simulation and Training, University of Central Florida, June 1993.

IMPLEMENTATION OF THE LASER MESSAGE PROTOCOL IN A DIS NETWORK

Randall K. Standridge
John D. Micheletti
Richard P. Weyrauch
Southwest Research Institute

ABSTRACT

This paper presents the results of the integration of the Deployable Forward Observer/Modular Universal Laser Equipment (DFO/MULE) training system into a multi-service Distributed Interactive Simulation (DIS) evaluation testbed, representing one of the first documented implementations of the Laser Protocol Data Unit (PDU). The DFO/MULE system provides target acquisition and tracking training for Artillery Forward Observers, Naval Gun Fire Spotters, and Forward Air Controllers as well as laser designation and rangefinding training. This stand-alone training system was modified to add a DIS networking capability, allowing ground-based Forward Observers to identify and designate targets for attack by artillery and aviation assets distributed within the Multi-service Distributed Training Testbed (MDTT) network. In addition to providing an overview of the system design and integration approach, this paper explores key issues which directly relate to implementation of the Laser PDU such as laser spot correlation with respect to terrain and targets, laser designation versus laser rangefinding, and laser-guided munitions modeling. The lessons learned from this implementation are discussed, along with suggestions and recommendations for future study and development.

ABOUT THE AUTHORS

Randall K. Standridge is a Senior Research Engineer in the Training Systems Development Section at Southwest Research Institute. He has over six years of experience in systems modeling, analog and digital design, and system integration and testing as applied to training system development. He received a B.S. and an M.S. in Electrical Engineering from the University of Texas at Arlington. **Address:** Southwest Research Institute, 6220 Culebra Road, San Antonio, Texas 78238. **Telephone:** (210) 522-3992.

John D. Micheletti is a Research Analyst in the Training Systems Development Section at Southwest Research Institute. He has over ten years of experience in the design, coding, testing, and documentation of avionics and simulation software systems. He received a B.A. in Computer Science from the University of Texas at Austin.

Richard P. Weyrauch is a Research Engineer in the Advanced Simulation and Training Concepts Section at Southwest Research Institute. He has over five years of experience in the design and development of simulation and visualization software for distributed processors. He received a B.S. in Electrical Engineering from the University of Wisconsin at Madison.

IMPLEMENTATION OF THE LASER MESSAGE PROTOCOL IN A DIS NETWORK

Randall K. Standridge
John D. Micheletti
Richard P. Weyrauch
Southwest Research Institute

INTRODUCTION

The Multi-Service Distributed Training Testbed (MDTT) is an ongoing Research and Development (R&D) project focused on assessing Distributed Interactive Simulation (DIS)-based training strategies and methods for multi-Service combat mission training. The Close Air Support mission was chosen due to the significant human performance challenges involved in integrating the various operational components. As illustrated in Figure 1, the testbed consists of distributed simulators representing aircraft (both CAS weapon delivery platforms and airborne Forward Air Controllers), manned ground vehicle simulators, computer-generated ground vehicles (opposing forces), and Dismounted Infantry (DI) functioning as a Forward Air Controller (FAC) element.

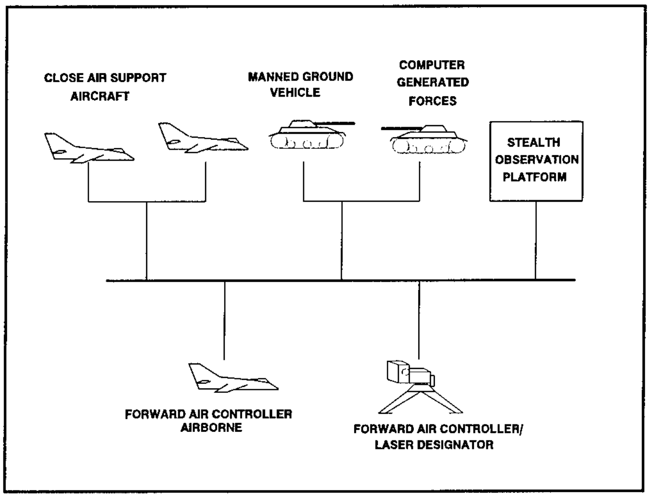


Figure 1 - Distributed Testbed Configuration

The Deployable Forward Observer/Modular Universal Laser Equipment (DFO/MULE) training system was chosen to participate in the testbed in order to provide a means for the ground-based FAC to observe and identify ground targets and engage them by controlling air assets. The DFO/MULE was originally developed as a stand-alone training system to provide target acquisition and tracking training for Artillery Forward Observers, Naval Gun Fire Spotters, and Forward Air Controllers. In addition, the DFO/MULE contains a simulated Laser Designator Rangefinder Module (LDRM) which allows the training of laser-guided munitions delivery using a laser designator.

SYSTEM OVERVIEW

Testbed Requirements

Integration of the DFO/MULE training system into the MDTT required that the stand-alone system be modified to meet the following general functional requirements:

1. All distributed simulations had to be interoperable as specified by the DIS standard 2.0 Draft 3.

2. The integrated DFO/MULE system had to support the Entity State, Detonate, Fire, Laser, Message, Event, Start/Resume, Stop Freeze, and Acknowledge Protocol Data Units.

3. The DFO/MULE system had to generate the Laser Protocol Data Unit (PDU) so that a Laser Guided Bomb (LGB) simulation (executed by the shooter) could realistically acquire and track the laser spot.

4. The laser spot position had to be modeled so that its location could be determined both with respect to entities within the field of view and with respect to the terrain.

5. Ground and air vehicle positions, velocity, orientation, and type had to be displayed based on Entity State PDUs received over the network.

6. Munition detonations had to be processed and blast effects displayed as appropriate.

DFO/MULE System

In order to understand the implementation issues involved in integrating the DFO/MULE training system into a distributed training environment, it is necessary to first understand the basic system design and associated limitations.

The DFO/MULE is a deployable, modular, personal computer (PC)-based system that provides training for MULE, Naval Gun Fire (NGF), Artillery (ARTY), and Close Air Support (CAS) personnel. Figure 2 illustrates the system configuration consisting of an Instructor Operator Station (IOS), a projector image-generator computer, a MULE image-generator computer, a high-resolution projector, and a simulated MULE. For DIS operation, the system was coupled with a government-furnished Network Interface Unit (NIU). In order to reduce the impact to the existing stand-alone DFO/MULE, the NIU was interfaced over the existing DFO/MULE ethernet local area network (LAN).

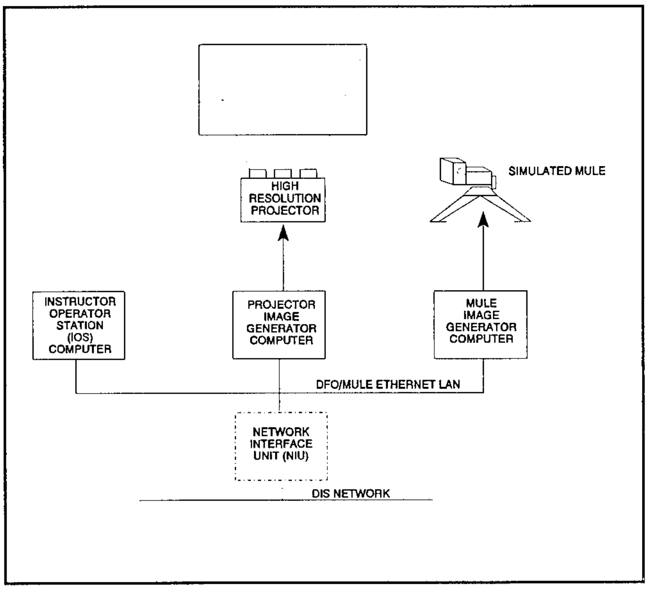


Figure 2 - DFO/MULE System Configuration

Under IOS control, the projector image generator displays either a 45-degree field of view (FOV) or a simulated binocular view with an 8-degree FOV. The MULE image generator produces a simulated Laser Designator Rangefinder Module (LDRM) daylight view (approximate 3.6-degree FOV) as well as a simulated thermal nightsight view (selectable 6-degree or 2-degree views). The background scenes used on all display devices are digitized high-resolution terrain images producing an extremely rich visual presentation. Terrain scene images are registered and correlated with elevation data derived from Digital Terrain Elevation Data (DTED) so that accurate visibility effects (occulting, pitch, etc.) are rendered for moving targets and munitions effects.

Ground vehicles, fixed and rotary wing aircraft, and munitions effects are generated by overlaying detailed graphical symbols scaled in size according to range and occulted by terrain and cultural features. Unlike higher-cost image-generation systems which utilize 3-dimensional (3-D) models of vehicles and terrain, the DFO/MULE system represents vehicles as a collection of 2-dimensional (2-D) images rendered from a 3-D model. Each set of images shows the vehicle in 198 different pitch and direction aspects. These images are called and displayed in real time to reflect the view of the vehicle from the observer's position. The detail of the target image is limited only by the 3-D model used to generate the target aspects, therefore allowing more detailed targets than would be possible on a comparable system generating targets in real time.

The display systems and the interaction of the user with the simulated targets are extremely important in any target designation system. The DFO/MULE provides two separate visual channels: the projected image (wide FOV and binocular FOV) and the simulated MULE (daysight and nightsight). The MULE images are produced on a 640x480 liquid crystal shutter monitor with separate reticle and focusing optics. The MULE images are correlated to the projected wide FOV image by means of a highly accurate set of calibrated position encoders. As the user acquires a target in the projected image and points the simulated MULE at the target on the screen, the image viewed in the MULE sight corresponds to that portion of the larger image.

IMPLEMENTATION ISSUES

Laser PDU

The Laser PDU was introduced to the DIS standard in order to provide a means of communicating lasing information in support of laser-guided munitions engagement. Although Version 2.0 Draft 4 of the DIS standard has changed this PDU name from Laser to Designator, the information contained in the message remains unchanged from Draft 3 which was used in the DFO/MULE modification. For consistency, the Laser PDU terminology will be used throughout the remainder of this paper. The Laser PDU contains the following information:

1. Standard PDU header including a time stamp

2. Identification of the entity generating the laser energy

3. Code name for the laser system being used

4. Identification of the entity being lased as determined by the lasing simulation

5. Laser code

6. Transmitted laser power in watts

7. Wavelength of laser energy in microns

8. Location of the laser spot with respect to the lased entity's coordinate system

9. Location of the laser spot in the World Coordinate System

The structure of this information within the Laser PDU is shown in Table 1. The message is transmitted at a rate of 10 times per second during any designation, with the last PDU transmitted being used to signal an inactive state by setting the laser power to zero.

The information contained in the various Laser PDU fields is generally allowed for a straightforward implementation. The intention of the laser code field is somewhat vague, and in fact required the only extension to the DIS standard implemented in the DFO/MULE modification. The DIS standard specifies an 8-bit enumeration for the laser code. Rather than using an arbitrary enumeration value, this field was implemented using the integer value corresponding to the LDRM Pulse Repetition Frequency (PRF) laser code. The LDRM code is in the range from 1111 (11.11 Hz) to 1788 (17.88 Hz) and is used to match the laser designator energy with a specific laser-guided munition. The 8-bit laser code field size was insufficient to specify an integer in the required range; therefore, the laser code field was extended to a 16-bit integer by using the 8-bit padding field shown in the PDU structure.

Table 1 - Laser PDU Structure

Laser PDU Fields	
PDU Header	Protocol version - 8-bit enumeration Exercise ID - 8-bit unsigned integer PDU Type - 8-bit enumeration Padding - 8-bits unused Time Stamp - 32-bit unsigned integer Length - 16-bit unsigned integer Padding - 16-bits unused
Lasing Entity ID	Site - 16-bit unsigned integer Application - 16-bit unsigned integer Entity - 16-bit unsigned integer
Code Name	16-bit enumeration
Laser Entity ID	Site - 16-bit unsigned integer Application - 16-bit unsigned integer Entity - 16-bit unsigned integer
Padding	8-bit unused
Laser Code	8-bit enumeration
Laser Power	32-bit floating point
Laser Wavelength	32-bit floating point
Laser Spot With Respect To Laser Entity	x-Coordinate - 32-bit floating point y-Coordinate - 32-bit floating point z-Coordinate - 32-bit floating point
Laser Spot Location	X-Coordinate - 64-bit floating point Y-Coordinate - 64-bit floating point Z-Coordinate - 64-bit floating point

Laser Spot Location

The primary mission of a laser designation system is to reflect laser energy from a target so that (1) laser-guided munitions can track on the laser signature for accurate delivery and/or (2) an aircraft equipped with a laser tracker system can use the designation to acquire and attack a target. Effective modeling of the spot location within the DFO/MULE was critical to its involvement in the MDTT program. Three elements of the laser were considered for implementation: laser spot diameter as a function of range, laser spot location in world coordinates, and laser spot location with respect to a lased entity's coordinate system.

The simulated laser spot location is determined by the interaction of the MULE operator with the target display. While designating a target, the operator attempts to keep the target within the reticle crosshairs of the MULE sight. Unlike the actual laser system, the resolution of the simulated laser spot location is limited by the resolution of the LDRM display system and position-sensing hardware, which were designed to allow for 1 pixel resolution in both the x and y dimensions. The area represented by each pixel in the display is therefore a function of the range from the simulated observation position. At a range of 1000 meters, 1 pixel represents a diameter of approximately 0.1 meter. At 4000 meters, that same 1 pixel represents a diameter of approximately 0.4 meters. The operational LDRM laser generates a divergent beam with a diameter approximated by the empirical formula:

Beam diameter in inches = 3 + 8 x (range/1000 m)

As illustrated in Figure 3, at moderate ranges (greater than 3000 meters) the laser spot area is a significant percentage of the area presented by a "typical" target. The beam area is closely approximated by a constant number of pixels independent of range. Based on this analysis, the laser spot position was implemented as a single pixel representing the center of the beam. The determination of whether a vehicle was lased was based on the position of this "targeting" pixel with respect to the target image. If the pixel was within the area of the target image, the target was classified as a lased entity. Conversely, if this pixel was not within the target image area, the beam was classified as off-target.

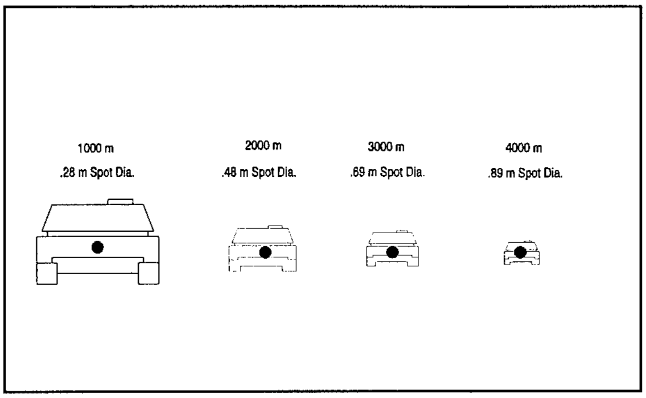


Figure 3 - Laser Spot Size Relative to Distance

In order to determine the location of the laser spot in world coordinates, the path of the laser was monitored and the projected intersection of the beam with the terrain was calculated. If the laser path did not intercept the location of a target entity, then the Laser PDU was transmitted with the terrain intersection point as the laser spot location. The transmitted coordinates were based on the beam intersection with the DFO/MULE terrain elevation database. Differences in elevation databases between simulators required that this spot location be clamped by other simulators to their local elevation based on the received X and Y location.

In the case where the DFO/MULE simulation calculates beam intersection with a target entity, it must supply the additional information regarding the laser spot location with respect to the entity's coordinate system. This presented an implementation challenge for two primary reasons, one specific to the DFO/MULE training system and the other a general issue for third-party target designation. The first problem in determining spot location involved calculating the 3-D location of a laser spot on a target that is essentially a 2-D image to the DFO/MULE. Unlike systems that use 3-D target models, the use of 2-D target images made determination of 3-D spot position extremely difficult within the constraints of the existing system. The second problem involved a more general issue of database correlation with regard to entity models. Just as differences in simulator terrain databases can produce correlation problems, differences in bounding volumes associated with target models can produce spot location problems. A laser spot location calculated by the designating simulator could potentially be inside or outside the bounding volume used by another simulator, requiring the spot location to be clamped to each system's representative bounding volume.

The spot location approach implemented in the DFO/MULE integration was based around a first-order determination of laser reflectivity. If the simulation determined that the laser aim point was within the displayed area of the target image, the Laser PDU was broadcast with the lased entity's location (entity coordinate system origin) as the location of the laser spot so that the location with respect to the lased entity was (0,0,0). This approach was dictated by several factors: the relatively large size of the laser spot with respect to the target bounding volumes, the limitations in the DFO/MULE 2-D target images, and the lack of higher-order laser reflectivity modeling for the LGB laser seeker; however, it was found to be quite satisfactory for the requirements of the testbed. The laser spot's relative visibility was then calculated by the LGB seeker model based on the empirical laser reflectivity model shown in Figure 4. The reflected laser energy was potentially visible to a seeker within a 140-degree "cone" centered along the laser line of incidence.

Implementation Summary

The following provides a summary of the issues addressed during the implementation of the Laser PDU within the context of DFO/MULE integration.

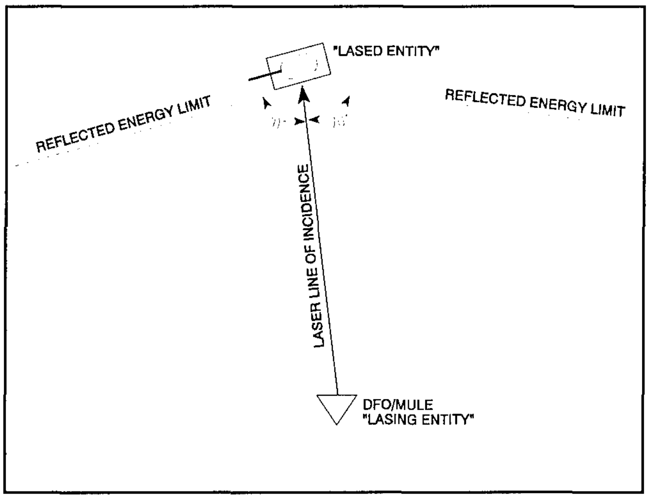


Figure 4 - Laser Energy Reflectivity Model

1. The Laser PDU was modified to make a 16-bit integer field for the laser code.

2. The transmitted coordinates of the laser spot with respect to a lased entity were given as the origin of the entity's coordinate system.

3. The number of pixels representing the laser spot diameter is essentially independent of the distance of the target from the laser position.

4. The laser spot intersection with the terrain was calculated and transmitted in the PDU when the laser spot was not incident on an entity.

5. A first-order reflected energy model was used by the LBG seeker simulation based on a 70-degree reflection "cone" on either side of the laser line of incidence.

LESSONS LEARNED

Overall Effectiveness

The implemented system was tested and found to successfully deliver simulated laser-guided munitions using third-party lasing. Quantitative validation of the total delivery system (e.g, laser system, aircraft platform, seeker model, bomb flight model, etc.) is beyond the intended scope of the testbed; however, feedback from subject matter experts indicates that the system adequately supports the testbed requirements for evaluating the effectiveness of Close Air Support training within a distributed environment.

Laser Spot Location

The single-pixel approach to determining the laser spot location provided inconsistent results. In certain cases, distant targets were unrealistically difficult to track with the simulated laser spot. This situation resulted from the "on-target" and "off-target" shifting of the center pixel used to calculate spot location. At a range of 3000 meters, a single pixel represents an area only 20 percent as large as an actual laser spot. The larger area of the actual spot allows energy to be reflected from the target even in situations where the center of the spot shifts off the target. By providing a larger bounding area for the laser spot, the spot tracking consistency can be significantly improved.

Network Bandwidth

Table 2 provides some network statistics during a 600-second period during which the DFO/MULE was lasing a target entity. Although the overall network traffic is relatively low, the Laser PDU accounts for 23% of the total DIS packet traffic during the analyzed period. This load is primarily a function of the fixed 10-Hz PDU broadcast required by the DIS standard. While the relatively small size of the testbed (three aircraft, 51 ground vehicles, and the DFO/MULE) did not present a problem with network load, larger exercises may well dictate approaches to save network bandwidth. As described by Evans (1993), the use of laser spot velocity and first-order dead reckoning may be an appropriate means of reducing the bandwidth required for designation broadcasts. Another approach might be to reduce the fixed-frequency update rate; however, further study is required to evaluate potential frequencies or dead-reckoning threshold limits.

Laser-Guided Weapon Flyout

The approach taken in implementing the Laser PDU in the MDTT program reflects the DIS standard, which states that the entity firing the laser-guided munition should simulate the guidance and detonation of the weapon after it is fired. This approach led to some simplification in laser spot location and reflected energy modeling which, although adequate for this and many other simulations, might not provide adequate performance in some cases. As proposed by Bouwens (1993) and others, once the weapon is fired, the laser designator essentially becomes the terminal controller. Weapon handover to the designator would allow the munition delivery to be tightly coupled to the designation function. Accurate energy reflectivity models and more precise laser spot positioning would result in potentially more accurate weapon delivery. However, additional development must be made in the protocols for transfer of entity control.

Table 2 - Sample Network Statistics

	Totals	Air	Ground	Comm	Weapons	Sim Mgmt	Emissions	Laser	Misc
Mean	36.73	6.50	14.90	3.16	1.60	0.04	1.82	8.43	0.11
Percent of Total	100%	18%	41%	9%	4%	0%	5%	23%	0%

600-second sample of DIS 2.0.3 packet data

CONCLUSION

This paper presented an overview of the modification of the DFO/MULE training system to make it compatible with a distributed simulation testbed. This modification is significant because it represents a successful implementation of the Laser PDU as specified in the DIS standard, and it documents the modification of a stand-alone training system into a DIS-compatible system.

ACKNOWLEDGEMENT

The authors wish to express their thanks to the Loral-Defense Systems Akron personnel at Armstrong Laboratory, Mesa, Arizona, for their efforts in DFO/MULE integration, and for permission to use network statistics compiled from the first evaluation of the MDTT program.

REFERENCES

Evans, A. B., "The DIS Laser PDU," Summary Report: Eighth Workshop on Standards for the Interoperability of Defense Simulations, Vol. I, IST-CR-93-44, March 1993.

Bouwens, C. L. "Strawman for Transfer of Entity Control," Summary Report: Eighth Workshop on Standards for the Interoperability of Defense Simulations, Vol. I, IST-CR-93-33, March 1993.

DYNAMIC ENVIRONMENT SIMULATION WITH DIS TECHNOLOGY

Mark Kilby, Curtis Lisle, Martin Altman, and Michelle Sartor

University of Central Florida Institute for Simulation and Training,

Orlando, FL

ABSTRACT

During the latest DIS workshops, the addition of a Terrain Manager or Environment Manager to DIS brought out understandable differences of opinion. Options discussed have covered the responsibilities of the Terrain and/or Environment Manager, its effect on the entities, and how entities keep track of the changing environment while considering whether any fundamental goals of the DIS paradigm such as "No central computer for event scheduling or conflict resolution" are violated. However, providing a consistent and dynamic environment in DIS exercises requires more than single environment management module, whether it be per network or per node. Instead, modifications to the simulation support architecture as a whole, must be contemplated.

Issues such as network bandwidth, CPU performance, and scalability must be considered by a system architecture that supports dynamic environments in a distributed interactive simulation. To address this need, several architectures are presented which could support dynamic entity/environment interaction. As each architecture is discussed, results and measurements made from prototype software are presented to point out strengths and weaknesses. IST demonstrated networked Dynamic Terrain (DT) capability using the most recent architecture at the I/ITSEC'93 conference. The architecture supported changes to the terrain profile, an extended DIS protocol, and provided a consistent way to manage changes made by entities.

ABOUT THE AUTHORS

Mark Kilby is a Visual Systems Scientist and principal investigator of IST's Dynamic Terrain project. Mr. Kilby received his BSEE and MSEE from the University of Florida

Curt Lisle is a Visual Systems Scientist at IST and analysis team leader on the Dynamic Terrain project. He received his BSEE from Georgia Institute of Technology and MS in Computer Science from the University of Central Florida (UCF). Also, Mr. Lisle is currently a Computer Science Ph.D. student at UCF.

Martin Altman is a Visual Systems Scientist at IST and architecture team leader on the Dynamic Terrain project. He received a BS in Computer Science from UCF, Summa Cum Laude.

Michelle Sartor is a Visual Systems Scientist at IST and real-time modeling team leader on the Dynamic Terrain project. She received a BS in Biology from Florida State University and a BSEE and MSEE from UCF.

INTRODUCTION

The purpose of a simulation support architecture which provides a Dynamic Environment within a Distributed Interactive Simulation (DIS) exercise is to enhance the effectiveness of the exercise. This enhancement is achieved by allowing entity simulators to affect and be affected by changes in the simulated environment. To provide this enhancement to the current DIS paradigm, several issues must be examined.

This paper presents issues that should be studied when defining a system-level architecture for Dynamic Environments in the DIS paradigm. First, concerns of the DIS community with respect to Dynamic Environments are discussed. Second, the objectives and assumptions for providing a Dynamic Environment within the DIS paradigm are presented. To meet these objectives and substantiate these assumptions, our research has explored a number of architectural options for Dynamic Environments within the DIS paradigm. These architectural prototypes and analytical results are presented next. The most recent prototype was demonstrated at the 1993 I/ITSEC conference. Lessons learned from this implementation are discussed. Finally, conclusions on results to date are presented.

Dynamic Environment Issues in the DIS Community

The past several DIS workshops have focused on two main issues concerning Dynamic Environments: a central environmental server and definition of the PDUs to transmit environmental changes during a DIS exercise.

Central Server or No Central Server. The concept of a Terrain Manager or Environment Manager in the DIS paradigm has generated understandable differences of opinion. The discussions in previous DIS workshops have covered the possible responsibilities of the Terrain and/or Environment Manager , its effect on the entities, and how entities keep track of the changing environment -- all the while considering whether this violates any fundamental goals of the DIS paradigm such as "No central computer for event scheduling or conflict resolution" (IST, 1993a).

Most of these discussions have focused on a single architecture which solves all the needs of the DIS community for a dynamic environment. In turn, much of the debate on this anticipated single architecture has centered around whether a central server is a necessary or desirable foundation of the architecture. A central server will have problems with throughput as the scenario scales to a larger number of entities. However, distributed architectures have problems with data redundancy, latency, and loose coupling between the CPUs on different simulators. The choice of options can not be resolved until dynamic environment requirements for a variety of possible applications in DIS are developed and prototypes to explore and meet these requirements are constructed.

Environment PDU Definitions. Much of the discussion on PDUs has assumed that PDUs will be sufficient to maintain temporal and spatial consistency between the diverse number of possible players in a DIS exercise and for the wide range of applications proposed for DIS (e.g., training, mission planning and rehearsal, doctrine development, virtual prototyping) (IST, 1993a). However, the supporting architecture must be considered as well. Furthermore, these PDUs need to support a variety of data requirements, both current and future. Each environmental PDU designed must consider the data requirements of the simulation platforms, the various environmental models, and the variety of applications proposed for DIS. Otherwise, unnecessary PDUs may be developed that add to the overall complexity and maintainability of DIS-compliant simulator hardware and software.

Objectives of a Dynamic Environment Architecture In DIS

To address these issues, objectives must be set for evaluating solutions to provide a dynamic, interactive environment for DIS exercises.

(O1) Sufficient Performance. Because the architecture must support several different types of simulations, it must support changes to the environment and provide these updates to all DIS nodes at a rate at least equivalent to the update rate of the fastest simulator in a particular DIS exercise. This does not mean transmitting PDUs at this rate. The current DIS standard provides mechanisms for reducing the communication of state changes of entities through dead reckoning mechanisms (IST, 1993). Also, performance is an important criteria, but not the most important. Focusing only on performance will deliver a short term solution which precludes other equally important objectives (listed below). These additional objectives could provide for the longevity and widespread use of the DIS standard.

(O2) Consistency of Representation. The architecture must also provide a uniform representation of the environment to all entity simulators and *a uniform methodology for accessing and changing the environment* to players in a DIS exercise. Thus, a standard mechanism must be developed to view and induce environmental changes by entities. The intent is to reduce correlation problems.

(O3) Scalability. One of the goals of the DIS standard is to support forces of varying sizes. Therefore, architectural solutions should work equally well for 10 or 10,000 entities. The architecture should support an increasing number of entities and their interaction with the environment through the addition of processing resources only. No change to the architecture design should be required.

(O4) Flexibility and Extendibility. Architectural solutions to provide a dynamic, interactive environment requires flexibility in a number of areas. First, the architecture should support multiple types of players whether they be man-in-the-loop simulators, constructive simulations, computer generated forces, or a combination of these. Second, multiple types of applications must be supported including training, testing, mission planning, and mission rehearsal (IST, 1993a). Also, the architecture must support multiple environment models and environmental effects models. An environment model refers to a simulation of naturally occurring phenomena while environmental effects refers to the effect of this phenomena on man-made devices or other natural phenomena. Therefore, rainfall would be provided by an environment model while effects on sensors and vehicle mobility would be provided through environmental effects models. These objectives for flexibility not only apply to current applications, but future applications as well. Thus, the system developed should accommodate future applications with minimal modifications.

(O5) Sufficient Realism. Different exercises as well as individual simulators within an exercise will require different levels of spatial and temporal fidelity. The architecture should support these multiple levels of fidelity at any resolution, size, or orientation.

Assumptions

To support the objectives listed previously, assumptions have been made in our research which have been derived through a number of sources (DIS, 1993, Lisle, et.al. 1994, E2DIS, 1993). Our research focuses on dynamic terrain, yet these assumptions are equally applicable to a complete dynamic environment.

(A1) Reconfigurability. It is unlikely that a single "golden" architecture will support all current and future DIS applications. Other authors have pointed out that the types of environment information required for a DIS exercise will vary depending upon the types of entities that are interacting and the goals of the exercise (Kamsickas, 1993).

This makes the choice between a central versus a distributed server unclear. A central server provides a consistent representation and reduces the processing power required of the individual nodes. However, the central server proves unscalable for a large number of entities and is not inherently robust (i.e., a single point of failure). A distributed server proves more scalable but comes at the cost of data redundancy, latency, and correlation complexity.

A hybrid system may be the best approach (Kamsickas, 1993). A single "golden architecture" which will solve all DIS dynamic environment requirements may be unfeasible, especially since there is a great variety of intended uses for DIS. Instead, different architectures for different applications may address the problem more effectively. For example, the central server will be cost-effective if it provides enough performance for a particular application having only a few entities. It may also prove to be the best solution for local collections of legacy simulators which cannot be easily modified. A fully-distributed system would be best for an application of DIS over a Wide-Area Network with low bandwidth in between DIS cells. A hybrid between centralized and distributed systems is an effective compromise for many applications. To support a variety of disparate needs with a single system and to address objectives O3 and O4, the system's architecture should be reconfigurable.

(A2) Unscripted changes to the Terrain Maintained Through an Active Database. Instead of maintaining a list of environment changing events, all changes will be incorporated into an active terrain database. Some image generator vendors are currently considering active databases as a modification to their current designs (Rich, 1992).

(A3) Applications Decoupled from Data Structures. To support objectives O3 and O4, the entity simulators should not depend on the structure of the data used by the simulation architecture to transmit and maintain a Dynamic Environment. These simulations should only depend on the content of the data. This can be provided through a standard interface, or query mechanism.

(A4) Arbitrary Number of Terrain Attributes. Typically, elevation and slope are the most frequently used attributes of the terrain. However, future applications will require such attributes as temperature, soil strength, and moisture content. Furthermore, different entity simulators need different attributes. For instance, a flight simulator with infrared sensors may only require terrain elevation, slope, and thermal attributes. However, a tank simulator would require additional terrain attributes to compute effects such as mobility.

(A5) Dynamic Update of Environment State. Just as Entity State PDUs transmit the change of an entity's state, terrain state PDUs should convey the change in the state of the environment.

(A6) Open System Design. Solutions to providing the Dynamic Environment within a DIS exercise should not be biased toward one vendor (e.g., a particular polygonization scheme).

(A7) Consistency with current vendor technology.

(A8) Support for Verification and Validation. The simulation architecture must provide a means to validate models and simulations in DIS exercises with minimal intrusion.

(A9) Support for Updates to Late Joining Entities. Players joining after the start of an exercise must be able to rapidly update their environmental database to match the current simulated environment.

(A10) Hierarchical Configurations. Certain exercises may require areas of terrain at different resolutions. Hierarchical approaches are well known solutions to this problem.

(A11) Geographic Segmentation. Allow for separate Terrain Managers to be responsible for geographically unique portions of the gaming area during an exercise.

(A12) Fault Tolerance. Due to the amount of resources that may be dedicated for a particular DIS exercise, the architecture should be minimally robust to prevent interruption due to minor errors.

(A13) Support Entity Simulator Queries for Environment Data both Locally and Remotely. Simulators should be provided with environmental data when required and not when it can be supplied by the architecture.

(A14) Assume and Relinquish Control of Steady State Objects. The Dynamic Environment support architecture could assume and relinquish control of steady-state objects (such as destroyed vehicles, bridges, buildings). It will also be capable of removing objects from the simulation as appropriate.[1]

(A15) Large Scale Scripted Environmental State Changes. The architecture should include the ability to replay scripted large scale environmental state changes to support simulator preprocessing of predistributed data. This will aid physically-accurate sensor simulations which currently require preprocessing to allow real-time performance. These predistributed databases can be replaced with models as they become available.

EVOLUTION OF A SOFTWARE ARCHITECTURE TO SUPPORT DYNAMIC TERRAIN

In seeking out architectures that will satisfy the objectives and qualify the assumptions listed previously, our research has explored a number of architecture options for providing a Dynamic Environment to DIS exercises. The variations of the architecture and analytical results are discussed below after a review of some initial design decisions.

[1]This issue is still under debate within the DIS Simulated Environment Working Group with some members being opposed to this assumption (Kamsickas, 1993).

Initial Design Decisions

To be consistent with current CIG technology (see A7), several design decisions were made.

2 1/2 D Representation. The portion of the terrain that is usually most relevant to a given scenario is the surface or "skin". This is sometimes referred to as a 2 1/2 dimensional representation.

No Multi-Valued Areas. An artifact of the 2 1/2 D representation, there is no concept of multi-valued areas (locations having more than one elevation value). Thus there is no generalized mechanism for representing caves or tunnels. Such things, if modeled, are considered as features or are otherwise handled as special cases.

Support Polygonal Based Representations. Most image generators are polygon based. In this representation, the terrain is essentially a large polyhedron. While research should explore alternative representations, we cannot ignore systems that are polygon based. It is also important to note that the different polygonization schemes have traditionally been a source of problems when linking heterogeneous simulators.

Gridded Source Data. The elevation source data for any given piece of terrain often comes in a gridded format where for each location there is one associated elevation value.

New Concepts in Environment Data Representation

To satisfy several objectives and assumptions (O2 - O5, A1 - A7) research was conducted in determining new data structures and data access methods for an environment that could support current polygonal-based training simulations as well as future applications. First, mathematical surfaces were determined to be the optimal choice in obtaining resolution independence. A terrain accuracy experiment using the 1992 I/ITSEC source database for the interoperability demonstration indicated that these data representations could represent the same terrain with fewer data points than equivalent gridded representations from which polygonal databases are currently constructed. Also, these surfaces were not subject to the errors of resampling at different orientations as are gridded representations (Lisle, et. al., 1994). Such representations may even produce very efficient environmental PDU designs.

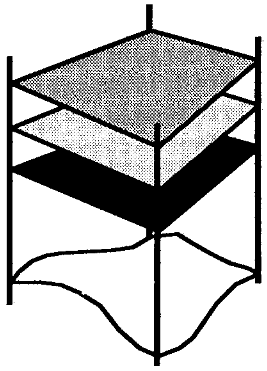


Figure 1: The Dynamic Terrain Database

To simplify the use of this new data structure by different simulators, a standard conceptual model was also developed. This conceptual model is referred to as the Dynamic Terrain Database (DTDB) (see Figure 1). Different attributes, including the terrain surface, are represented as layers within the database with each layer being represented by a mathematical surface. To access the data of an attribute layer, a standard interface to the database is provided in the form of an arbitrary quadrilateral area. That is, a simulator can request data in the form of gridded area of any size, resolution, or orientation. From this gridded data, a polygonized terrain surface can be easily generated.

Architecture 1 - Central Server

This first architecture (see Figure 2) was designed to serve two main purposes. First, the initial design was to address the fundamental problems in Dynamic Terrain (see O1 - O5 and A2 - A7). Second, the design should help determine how DIS could benefit from existing academic research.

Definitions. The following terms are used in describing Architecture 1:

Dynamic Terrain Portal (DTP): The DTP is the only part of the system with a connection to both the DIS network and the internal Dynamic Terrain architecture. It serves as the interface to and from the DIS network. Its responsibilities can be grouped into three main categories. First, inbound task responsibilities include watching the DIS network for terrain changing events (e.g. bulldozer digging, detonations on terrain).

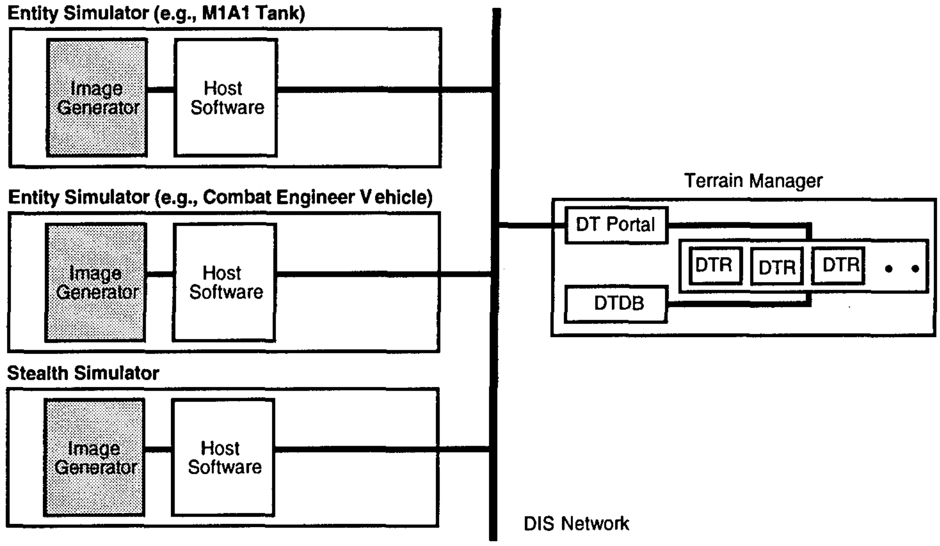


Figure 2: Architecture 1

Second, internal task responsibilities include handling details of the DT Simulation (e.g. Remote Entity Approximations, interaction between the DTDB and DTRs). Finally, outbound task responsibilities include transmitting terrain changes to simulators via new DIS messages.

Dynamic Terrain Resource (DTR): A specific physics-based algorithm is contained inside each DTR. The DT architecture allows for DTRs to be added as they are designed.

Dynamic Terrain Database (DTDB): This object maintains the simulation's environment database. Ground elevations and terrain attributes are stored here with any information necessary for the function of the DTRs.

Analytical Results. A software simulation of this system was developed to evaluate system behavior against DIS PDU streams. The following is a brief summary of the lessons learned from this experience:

• Data transfer traffic between DTRs, the portal, and the database was heavy. These functions should be on the same CPU or a shared memory multiprocessor.

• The design of the database was more critical to overall system performance than was originally expected, particularly in the area of record-locking and its affects on timing of multiple jobs simultaneously processed by the Terrain Manager.

• The DTP became a bottleneck since it is the only point of connection between the DIS network and the DTRs.

• This architecture was not sufficiently scalable because of access conflicts to the single, shared Database resource and the traffic load through the DT Portal.

Summary. It was understood before the development of Architecture 1, that this design did not completely fulfill the objectives presented earlier. However, it was conceived as a step in the process of requirement definition and provided useful experimental results on the central server approach.

Architecture 2 - Distributed Server

Architecture 2 (see Figure 3) is based on Architecture 1 with more detail focused on the Terrain Manager implementation (Horan et. al., 1993). Development began with several goals:

• Develop details of the Terrain Manager design. Look for efficiency increases and any major problems of the original architecture design (e.g., the DTP bottleneck).

• Expand the parallelism of the Terrain Manager. Look at the component responsibilities and how they could be allocated among a varying number of parallel processors.

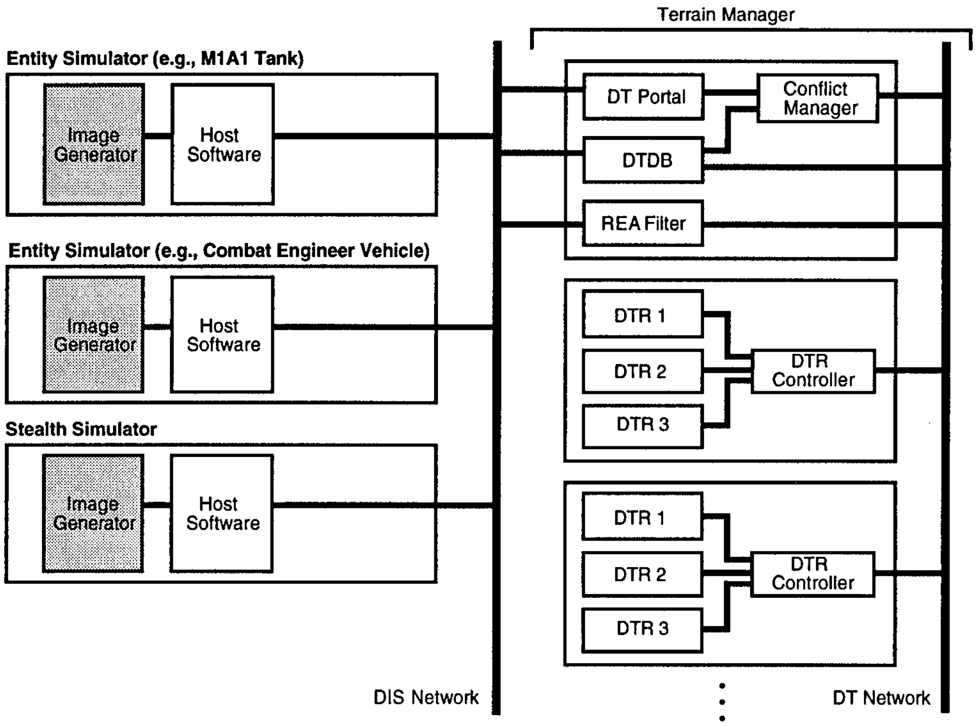


Figure 3: Architecture 2

• Support dynamic load balancing among multiple processors to consolidate all physical modeling jobs working on a specific geographic area to one processor. This permits sharing of a local database cache by the DTRs.

Additional components were defined in Architecture 2 to improve functionality. First, the functionality of the DT Portal (DTP) was split between the DTP, the Conflict Manager, the Remote Entity Approximation (REA) Filter, and the DTR Controllers. In this new design, the DTP monitors the DIS PDU stream for environment (i.e., terrain) changing events. When one is detected, the DTP determines the physical model invoked, assigns a priority to this task, then hands the task over to the Conflict Manager. The Conflict Manager, in turn, determines the machine on which to run the physical model based on available processing resources. The Conflict Manager hands the tasks to a specific DTR Controller, a component which controls all DTRs on a specific machine. The Conflict Manager also receives changes to the DTDB from a DTR and resolves conflicts when two or more DTRs wish to modify the same location in the database. Once the DTR Controller receives a task, the task is then passed to a DTR to run the physical model and compute the change to the environment. Also, the REA filter is used by the DTRs to monitor an environment-changing event once it has been identified by the DTP. The DTDB, while similar to the version in Architecture 1, now has access to the network for directly sending out changes to the database via PDUs. Thus, the communications responsibilities of the DTP are spread across the DTP, REA Filter, and the DTDB and reduces the communications bottleneck problem.

Analytical Results. A software simulation of this architecture provided significant lessons. Since the DTP was simplified, its main purpose became management of the processing jobs within the Terrain Manager. However, even with a dedicated dead-reckoning input path (i.e., the REA Filter), network I/O was still a limiting factor. With this design, the limit was not between the Terrain Manager and the DIS network, but within the manager itself. The DTDB serialized transactions from multiple DTRs by queueing requests. Thus, the parallel processors and the heuristics developed to locate and migrate jobs according to the CPU assignment of the job and the CPU load were not fully exercised.

Summary. This architecture was more scalable than the first. However, the limitations of a single database showed up more clearly. This architecture also proved more fault tolerant

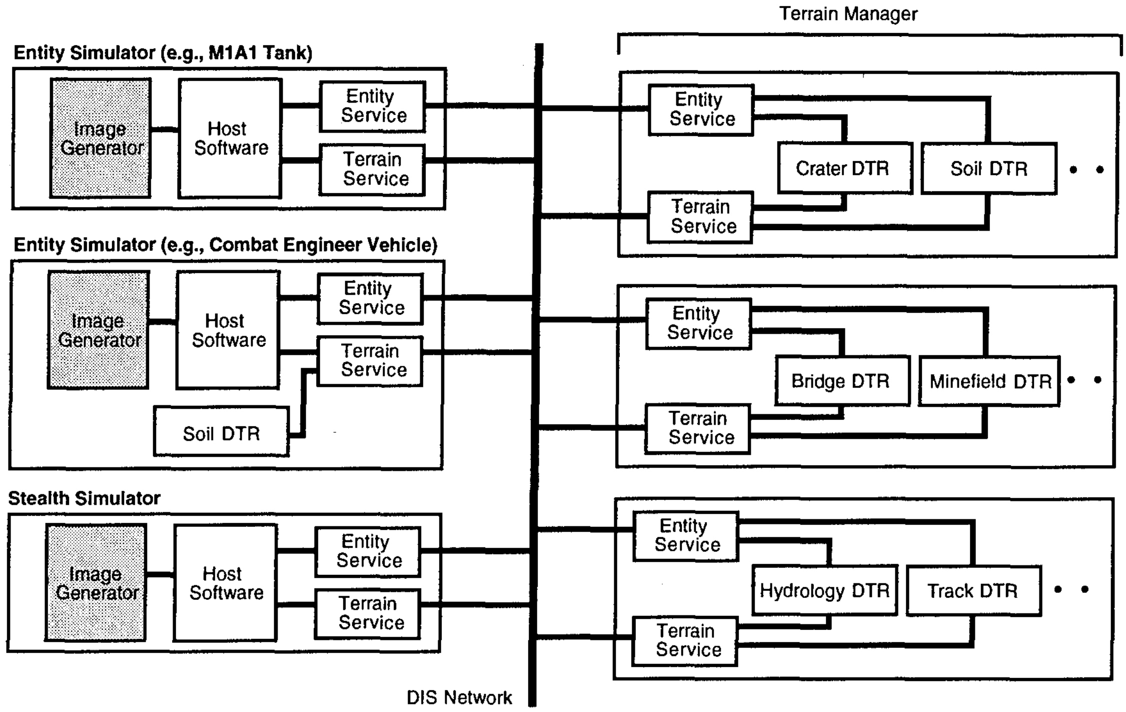


Figure 4: Architecture 3

(Horan, 1993). Failures of any of the DTR Controllers or DTRs were easy to recover. However the DTP and the REA Filter still required redundancy if the system was to remain tolerant of hardware failures.

Architecture #3 - Polymorphic Simulation Architecture

At this point in the project, several useful system designs had been developed, but a system design was still required which met the original objectives. A different approach was built based on the assumption of utility processes running in the background of all simulation host computers in the DIS exercise. The resulting shared environment was implemented through the interaction of these utility programs, called services. Using a Client/Server paradigm, the services are the servers while the simulation application programs are the clients. The idea is for the services to provide a common interface to the dynamic, shared environment for both entity simulation platforms that cause environmental changes and physics-based environment models which determine those environment changes (see Figure 4). The services make up the simulation support architecture which maintains a consistent environmental representation and isolates the environmental models and entity simulators from effects due to future design changes in this architecture.

To determine the form of this common interface, common data requirements of entity simulations and environmental models were determined. With a focus on Dynamic Terrain, these clients require constant updates on two types of data: terrain and entities in the DIS exercise. Thus, two types of services were developed. One is referred to as the Entity Service and the second is the Terrain Service.

Entity Service. One of the requirements of the DIS environment is that each node maintains representations of other entities (ghosts, or remote entity approximations) within its area of interest. Entity State PDUs are received from the DIS network and dead reckoning is performed on these approximations. In an environment where there is only one application per node (i.e., physical machine), it matters little where this functionality resides. On the other hand, if there were the capability to have several applications residing on one node (e.g., entity simulators and environmental models), the location of this functionality becomes important.

Consider the capability of several applications resident on one node. Many of the components of the DT architecture (e.g., DTP and DTRs) are individually small in resource processing requirements and do not warrant allocation of an entire machine. This leads to two approaches.

These small components can be bundled into one large, eclectic application or they could reside in separate applications. The issue then becomes one of where to place the entity tracking capability.

If grouped in one large application, the entity tracking functionality is trivially included with the DT architecture components as modules within the single application. Yet, both development and maintenance are unnecessarily complicated. If the components are isolated in several small applications on the same machine, the development and maintenance problems are reduced. Yet, location of the entity tracking functionality becomes a problem. Duplicating this functionality in each application produces inefficiency and consistency problems.

This reasoning results in a need for a single application that can perform DIS communication, dead reckon remote entities, and support simultaneous client applications that require subsets of this entity information, all running at different frame rates. This application is referred to as the Entity Service (see Figure 5).

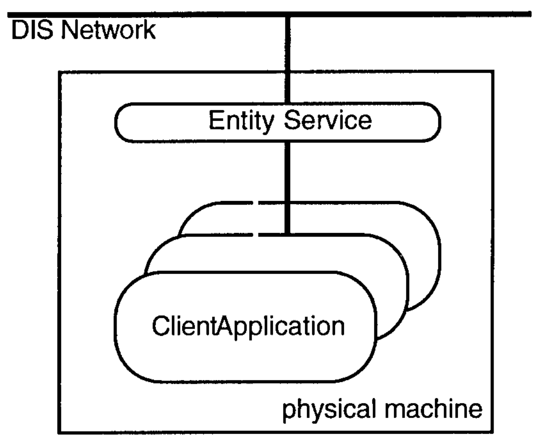


Figure 5: Entity Service

The Entity Service runs as a separate application and serves as the intermediary between the client applications and the DIS network. There is one copy of the Entity Service per machine, which handles Entity State, Fire and Detonation PDUs. Each application is granted a private channel for communication with the service. This decouples applications from one another, and allows for applications running at different frame rates to access the same service.

The functionalities of DIS communication and dead reckoning are encapsulated within the Entity Service. Each client application can safely assume that the most current entity state information is available from the service. The capability to support a small, arbitrary number of applications on the same machine is also gained. This reduces development and maintenance effort of the entire DIS simulation architecture by minimizing the "ripple effect" of design changes in one part of the architecture affecting other parts.

Terrain Service. In a fashion similar to the development of the Entity Service, a Terrain Service has been created to provide a consistent, common interface to the state of the terrain. The Terrain Service is also a separate application and contains the Dynamic Terrain Database described previously. It serves as intermediary between the DTDB and the client applications for queries and updates, as well as handling transmission and reception of environmental (i.e., terrain) PDUs. In addition, an experimental protocol was established for the Terrain Service to notify the client applications upon receipt of a terrain change. After notification, the client application must determine if it is affected by this change and needs to request the change from the Terrain Service (see A13).

Analytical Results. Currently, this architecture is under study and analytical results are not available at this time. However, this architecture was demonstrated at the 15th Interservice / Industry Training Systems and Education Conference 1993 (I/ITSEC'93). The demonstration showed *unscripted* changes to the gaming area database due to detonation of munitions and digging of anti-tank ditches and berms. Demonstration hardware consisted of several Silicon Graphics (SGI) workstations and an ESIG-2000 computer image generator on a local DIS network. SGI Indigo workstations operated various components of the Dynamic Terrain Architecture as well as simulators for bulldozers, an AVLB, and artillery rounds. An SGI Onyx and the ESIG-2000 both functioned as stealth platforms capable of simultaneously viewing all Dynamic Terrain changes occurring during the DIS simulation. The Onyx used IST-developed visualization software. Additional assistance was provided by Loral Advanced Distributed Simulation. Providing a modified GT120 image generator and using IST's dynamic soil model DTR, Loral demonstrated how this technology might be used to change SIMNET terrain databases in real-time. This demonstration achieved our project goal of exploring a means of integrating new DT technology into both current visual systems and future simulation systems (see O4, A6, and A7).

Summary. Central server, fully distributed, and hybrid configurations can be realized by Architecture 3. More research is required to determine which configuration should be used for a particular exercise. It is important to note that client applications are insulated from changes to the configuration. This benefit is gained by separating the terrain data structures from the client applications (i.e., entity simulators). Also, the software implementations of physical modeling algorithms were easier to develop using the services. Next, since access to environment state goes through an extra level of software indirection, flexibility is achieved at the cost of some performance. However, this performance can be regained by optimization. Finally, load balancing or associated database conflict resolution management has not been implemented at the time of this writing. We expect the flexibility provided by the Terrain Service to make the architecture tolerant of changes in this area.

SUMMARY

This paper has addressed several issues that must be considered when addressing a system-level architecture for Dynamic Environments in DIS. These issues are as follows:

• Environment state can be considered in a manner consistent with vehicle state. Occasional broadcasts and local dead reckoning can be used to minimize consumption of network bandwidth. This is a natural extension of the DIS protocol.

• The diverse needs for Dynamic Environments in DIS exercises can be addressed by a reconfigurable simulation support architecture. The Client/Server approach is one means to achieve this and provides a consistent common interface to the simulated environment.

The results and recommendations included in this paper evolved from a detailed study of the problem of Dynamic Terrain in DIS. However, many problems remain unsolved and should be addressed by continued research.

ACKNOWLEDGMENTS

This work is funded by the US Army Simulation Training and Instrumentation Command (STRICOM) under contract N61339-92-K-0001.

REFERENCES

E2DIS (1993). E2DIS Architecture Task meeting minutes. August 1993 and November 1993.

Horan W., Smith M., Altman M., Lisle C. (1993), “An Object-Oriented Environmental Server for DIS”, in Proceeding of the 9th Workshop on Standards for the Interoperability of Defense Simulations, IST-CR-93-39.2. September 1993. Institute for Simulation and Training, Orlando, FL. pp. 15-23.

Horan, W. (1993), “Scheduling in a Distributed Application to Support Environmental Servers for DIS (Distributed Interactive Simulation) Exercises”, Thesis Report, College of Engineering, University of Central Florida, December 1993.

IST (1993a). DIS Operational Concept 2.3, IST-93-25, Institute for Simulation and Training, Orlando, FL.

IST (1993b). Land Subgroup meeting minutes, in Proceeding of the 9th Workshop on Standards for the Interoperability of Defense Simulations, IST-CR-93-39.2. September 1993. Institute for Simulation and Training, Orlando, FL. pp. 43-86.

Kamsickas, G. M. (1993) "Distributed Simulation: Does Simulation Interoperability Need an Environment Server?", in Proceedings of the 15th Interservice/Industry Training Simulation and Education Conference (I/ITSEC). November 1993. Orlando, FL. pp. 235-244.

Lisle, C., Altman, M., Kilby, M., Sartor, M. (1994) "Architectures for Dynamic Terrain and Dynamic Environments in Distributed Interactive Simulation", in Proceeding of the 10th Workshop on Standards for the Interoperability of Defense Simulations. March 1994. Institute for Simulation and Training, Orlando, FL.

Rich, H. H. (1992). "The Active Database - Using Software to Save CIG Hardware", in Proceedings of the 14th Interservice/Industry Training Simulation and Education Conference (I/ITSEC). November 1992. Orlando, FL. pp. 858-866.

DEPLOYABLE ELECTRONIC COMBAT MISSION REHEARSAL, TRAINING AND PERFORMANCE SUPPORT

Patrick G. Heffernan & David W. Galloway
TRW Avionics & Surveillance Group, Warner Robins Avionics Laboratory
Warner Robins, Georgia 31088

ABSTRACT

This paper presents the approach and results of an internally funded project at TRW to develop a portable, self-contained electronic combat (EC) simulation system. This system, known as the Portable Electronic Combat Simulation (PECS) system, provides the ability to conduct EC mission rehearsal, part-task training, and performance support functions in a deployed state using one stand-alone package. For mission rehearsal and part-task training, this tool provides a real-time simulation of the threat environment, a high-fidelity aircraft flight path generator, an electronic warfare (EW) defensive systems processing and environment interaction, a countermeasures effectiveness model, and an audio and video interface to the user via a graphical user interface. For performance support functions, the system provides an encyclopedia of threat information and a tool for conducting initial and refresher training for specific EW defensive systems.

The PECS system real-time and off-line software is hosted on a single VME chassis and employs multiple 68030 and SPARC CPUs. The real-time simulations software was developed in a building block style allowing the user to rapidly reconfigure his EW defensive systems suite from the models available. The off-line software includes a toolset of editors to build mission files and the performance support functions.

This development effort demonstrated that effective real-time EC mission rehearsal and training and off-line performance support could be employed without large weapon system or aircraft part-task trainers. The PECS system software architecture also illustrated tremendous flexibility in supporting a number of different EW configurations, allowing new and qualified air crew members, from several different airframes, to learn and practice on a single turnkey system. The performance support function shows that air crew members can improve their EW knowledge base without the formal constraints of a CBT system.

ABOUT THE AUTHORS

Pat Heffernan is a senior software engineer at TRW. Mr. Heffernan has nine years of experience in the electronic warfare field. He flew operationally as an Electronic Warfare Officer aboard the MC-130E aircraft. He currently designs and develops simulation and training systems. Mr. Heffernan received a B.S. from the USAF Academy and another B.S. from the University of West Florida. (912.929.7406)

David W. Galloway is a senior systems engineer at TRW. Mr. Galloway has designed and developed real-time digital signal processing and data acquisition/control hardware and software for several radar applications and served as lead engineer during the development of the Integrated Electronic Combat Simulation System (IECSS). He is currently the hardware lead for a team implementing a software support facility for an F-15E avionics suite. Mr. Galloway received a M.S.E.E. and B.E.E. from Auburn University.

DEPLOYABLE ELECTRONIC COMBAT MISSION REHEARSAL, TRAINING AND PERFORMANCE SUPPORT

Patrick G. Heffernan, David W. Galloway
TRW Avionics & Surveillance Group, Warner Robins Avionics Laboratory
Warner Robins, Georgia

GENERAL DESCRIPTION

The portable electronic combat simulation (PECS) system was an internal development project at TRW to design, build, and demonstrate a deployable system capable of performing mission rehearsal and training functions for air crew members in the specialized area of electronic combat (EC) using commercial-off-the-shelf (COTS) hardware and software components. Due to the nature of large weapon system trainers (WSTs) and part task trainers (PTTs) and the integration of electronic combat tasks within these WSTs/PTTs, deployed air crew EC mission rehearsal and part-task training using these types of tools is almost non-existent. The objective of this project was to develop a proof of concept system that would provide the capability to provide real-time, high-fidelity electronic combat simulation interfaced with a graphical user interface (GUI) for mission rehearsal and training within the framework of a cost-effective, portable package.

The PECS system provides a full real-time EC simulation capability for multiple aircraft simulation models in either a networked or standalone configuration. That is, the PECS system can be networked to other PECS systems to simulate multiple aircraft if a training scenario involves other air crews or it can be operated as a standalone entity if the user requirements are for a single aircraft platform. It also has the capability to interface with dissimilar systems through the distributed interactive simulation (DIS) protocols.

As the system matured through the development stages, a performance support package allowing users to refresh and upgrade their knowledge on EW equipment, threats, and EW concepts was added. An off-line support system is hosted on the PECS to maintain threat and EW equipment databases and build mission scenarios for the real-time EC simulation. Currently, the PECS system is tailored for specific special operations aircraft

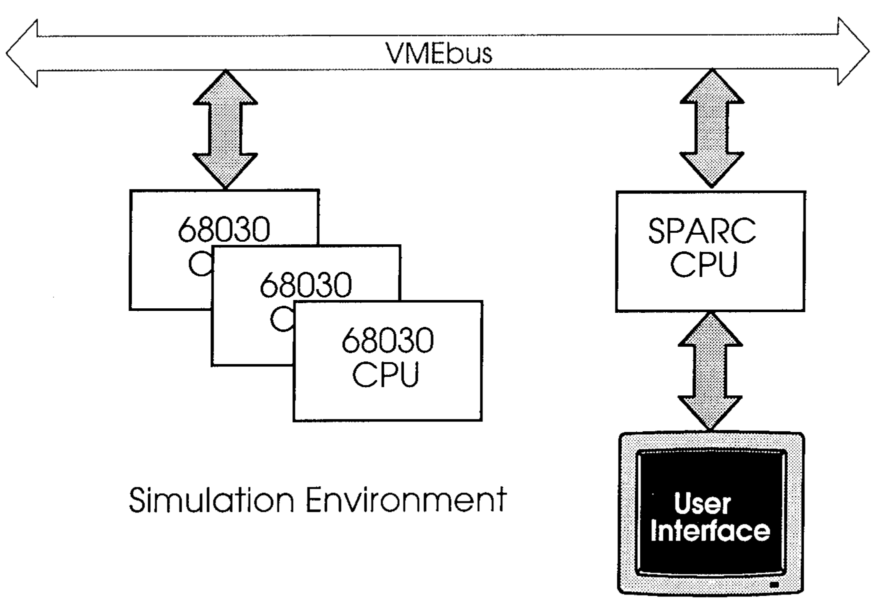


Figure 1. PECS Hardware Block Diagram

(MH-53J, MH-60G, and HC-130P) EW configurations, since many of the real-time simulation models were previously developed for the USAF.

HARDWARE & SOFTWARE ARCHITECTURE

The basic hardware architecture of the PECS system, as shown in Figure 1, uses a Force™ Target 32 system as the foundation. This system consists of a 19 inch desktop chassis, a 20 slot VMEbus based motherboard, and power supply with cooling fans. The chassis back panels have been modified to allow cable plug-ins in the rear of the unit. This chassis system was selected due to its lightweight nature, portability, and ability for growth. The VMEbus implementation was selected because it offered a wider and more powerful range of processing devices over a PC based implementation and at a more moderate cost than a minicomputer implementation.

The real-time software simulation executes on three general purpose VMEbus based single board computers. CPU1 is a Force™ CPU-30BE/16 (Motorola 68030 CPU) single board computer and it acts as the VMEbus master. CPU1 also provides the interface to the SCSI disk device containing the required data files for the real-time simulation and the interface to the Local Area Network (LAN) via ethernet. CPUs 2 and 3 are Force™ CPU-33B/4 (Motorola 68030 CPUs) single board computers running in a slave mode. These boards provide the processing capability for executing the high-fidelity simulation models required for real-time EC mission rehearsal and part-task training.

The GUI, off-line support system, and performance support system software execute on a Force™ SPARC CPU-2CE (Weitek W8701 SPARC CPU) single board computer utilizing the standard Sun™ UNIX™ operating system environment. The CPU-2CE also interfaces with dual SCSI disk devices to access the off-line executables and data files necessary to operate the system. A 19 inch color video monitor, dual audio amplified speakers, microphone, keyboard, mouse, and any required ethernet lines are connected to the SPARCboard via the back panel cable plug-ins.

The software architecture of the PECS system is based upon a series of modular software components as shown in Figure 2. The real-time software simulations executing on CPUs 1, 2, and 3 operate under Integrated Systems pSOS+™ real-time operating system. pSOS+™ allows the simulation's real-time executive software to transparently interface with the system hardware components to accomplish tasks like disk accesses, serial communications, and software interrupts.

The real-time simulation software is coded in a

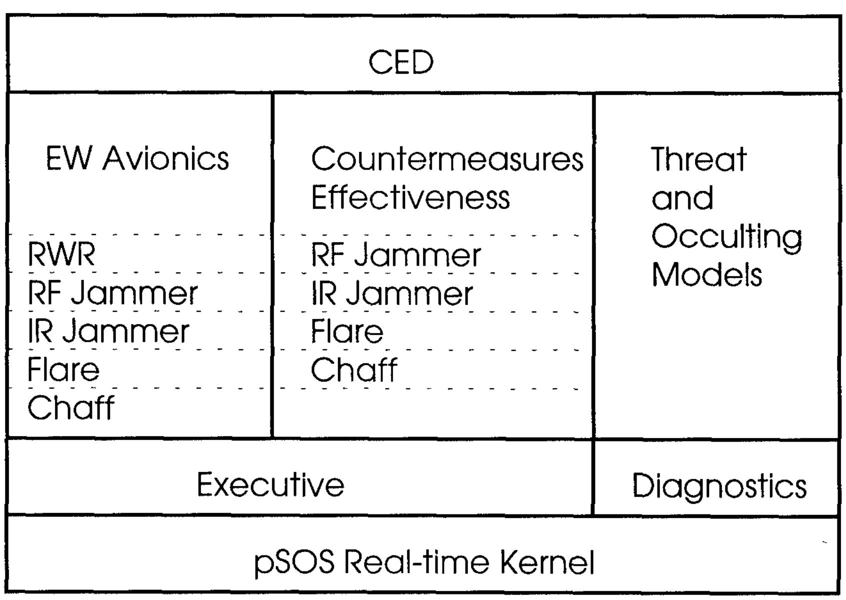


Figure 2. PECS Software Building Blocks.

combination of FORTRAN and C. These high order software languages were chosen due to the number of existing threat and EW avionics software modules. The real-time models are divided into the following major categories:

1) Threat and occulting models
2) EW avionics models,
3) Countermeasures effectiveness models
4) Executive software

The basic data communication path between the threat model simulation and the EW avionics models simulation is a shared memory area mapped to one of the simulation CPUs memory. This shared memory area, called the Common Environment Data (CED) data structure, contains all the information required to define the current electronic combat environment.

The CED describes the battlefield EC environment at any instant in time to the level of fidelity required for a training simulator. This EC environment is composed of data elements that describe threat and emitter entities, weapon entities, aircraft entities, infrared (IR) and electromagnetic (EM) jamming entities, and chaff and flare entities. The CED can be memory mapped so that networked PECS systems can access it across shared memory network using reflective memory techniques.

In a networked simulation, one of the PECS systems is declared as the master of the entire threat simulation environment. It is the master's responsibility to assimilate all scenario and aircraft information from the slave PECS systems to derive the information contained in the CED. That is, the slave PECS systems conduct processing on a number of simulation model functions and then they pass the information to the master PECS system which collects, processes, and redistributes the data necessary for scenario generation.

THREAT & EC SIMULATION HIGHLIGHTS

The PECS system provides a complete, high fidelity threat and EC simulation capability. Since many of the simulation system's models were developed from previous programs (Galloway, et al.; 1993), only a discussion of significant changes and important model highlights of the threat and EC simulation will be presented here. The PECS executes a closed looped EC simulation which integrates the three major components involved in an electronic combat simulation: threat, EW avionics, and countermeasures effectiveness.

The threat simulation includes models which activate and engage targets in a rule-based fashion, perform target assignments, utilize C3 information, drive computer controlled airborne interceptors, fire weapons, conduct 6 DOF weapon flyouts, and perform damage assessments. The threat models use threat and atmospheric data stored in the off-line database to accurately model threat weapon systems and emitters in the gaming environment. During real time execution, the threat simulation supports a maximum of 64 active threats, 128 active emitters, 8 active missiles, 8 active AAA bursts, and 8 active computer controlled airborne interceptors at any given instant in time. In addition, the threat simulation integrates with modules which perform threat encounter recordings, on-line modification of threats and threat status, and threat occulting (terrain masking functions). The threat occulting is performed by a child task on the SPARC board which accesses terrain elevation data stored on the SCSI disks. The occulting function obtains positional information from the threats and targets via the VMEbus, processes it, and passes the occulting results back to the threat simulation on the VMEbus. This process is repeated at a resolution required to support the threat simulation fidelity requirements.

The EW avionics simulation includes models which simulate the operational flight programs (OFPs), emitter identification databases (EIDs), and user displays and interfaces of systems which detect, identify, track, display, and jam or deceive threat systems. In the PECS system, the models drive the graphical user interface buttons, knobs, lights, and switches. The EW avionics OFP and EID simulation samples the threat environment, reads the threat parametric data, processes the threat data, displays the appropriate threat symbology, and creates the associated threat or verbal audio queues. The EW avionics simulation integrates with modules which enable on-line modification of model states (malfunctions) and enables countermeasures effectiveness analysis and simulation.

The countermeasures effectiveness simulation includes models which use the best available effectiveness data to perform a Monte Carlo simulation of jamming and expendable effects on the threat models. The active countermeasures systems models use positional and atmospheric information along with effectivity factor tables associated with jamming techniques and emitter parametrics to assess effectiveness. Expendable models use positional, atmospheric, and expendable timing and quantities to assess effectiveness. Jammer-to-signal calculations are performed and compared with target radar cross section (RCS) and IR signatures. The PECS system can have 8 active chaff clouds per system, 8 active flares per system, 1 active IR countermeasure per system, and 12 active RF countermeasures per system active at any instant in time.

REAL-TIME USER INTERFACE

The real-time graphical user interface was implemented using the Altia™ Design animated visual interface design tool. The software developer can use Altia™ Design to construct, animate, and stimulate static models and provide a seamless interface to the simulation model code. In the PECS system application, Altia™ Design was used to build the EW avionics air crew interfaces, informational data screens, user simulation control screens, and mission situational displays.

The basic structure of the GUI is shown in Figure 3. The user interface provides the actual windowed control environment to the system operator. This portion of the design was implemented using the Altia™ Design development tool. The client application acts as the interface or data conversion code between the user interface and the actual training simulation code. The interface between the client application and the threat and EW simulations is a shared memory area mapped in VME address space to be accessible by all the processors in the system.

The Altia™ Design tool presents the developer with a work space that Altia™ refers to as the universe. This work area is basically a blank area for drawing or producing the views that the end user requires to meet the training strategies. The universe acts as a palette for the client application developer to utilize in presenting the PECS operator with a clipped version of the universe. It is clipped in the sense that only a portion of the universe is presented to the operator in any one window. What portion of the universe that is displayed during run-time is

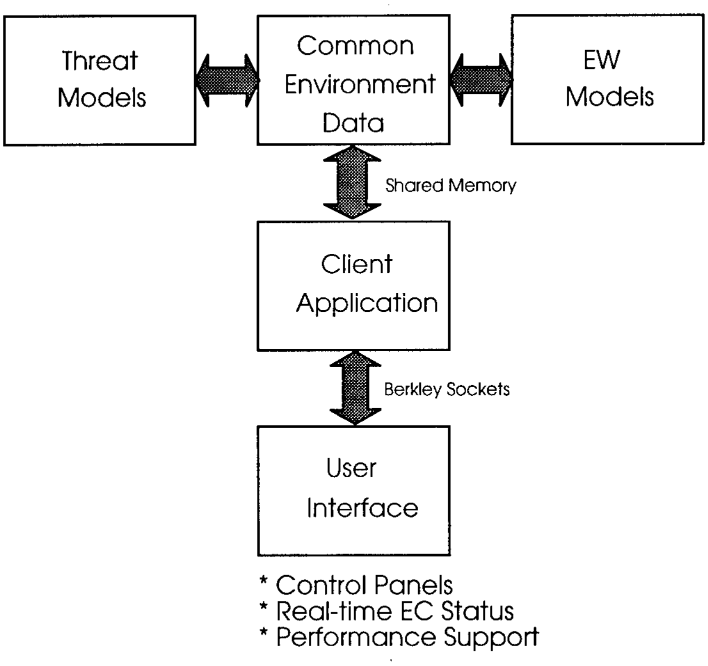


Figure 3. Altia Run Time Configuration.

preprogrammed by the client application developer.

The client application provides more than the view manipulation of the universe. The client application also responds to the run-time interrupts either generated by a stimulus from a displayed view or generated by a time event. An example of a stimulus interrupt would be the operator "pressing" a button on an RWR control panel. The event and the status of the button after the event is transmitted to the client application using operating system sockets. In this example, the client application would translate the button event into the data format defined by the CED documentation.

The combination of the user interface and the client application presents the operator with enough control capability to activate models, control model operation, and display simulation status from a mouse driven interface.

Upon startup, the operator is presented a simulation control panel utilizing one of the views into the Altia universe. The simulation control panel allows the user control of high level simulation functions like the simulation startup and freeze capability. The simulation control panel also presents a menu panel for the user to select the desired control panel views to support his current training session.

The PECS system client application program is coded in C and uses Altia™ Design library routines to manipulate the graphical views. The library routines allow the user to connect the client application to the Altia™ interface, register and process callbacks, create and manipulate graphical views, create and manipulate clone objects, and integrate Motif widgets as an Altia™ interface view.

In the PECS system application, for example, a generic radar warning receiver (RWR) threat display unit was created graphically using the Altia™ Design tool. The RWR screen, buttons, and knobs are drawn and then animated. The stimulus for the animated RWR objects are then defined as being either mouse or keyboard prompted.

Usually, the client application program registers a procedure or routine as a callback for an Altia™ interface event. After this is accomplished, the client application program waits for an event to occur. When the user pushes a button or turns a knob, the client application procedure recognizes the event through the event callback function. Similarly, information from the real-time simulation model is obtained by the client application program and passed to the GUI via the domain socket. The client application procedure is executed after a user specified time interval has elapsed. The time intervals are defined using an Altia™ timer event callback function. The procedure registered as a callback can receive the event data through the callback event timer. This allows data to be displayed to the user in real-time.

Figure 4 is a picture of the PECS system displaying a representative group of user interface screens.

OFF-LINE USER INTERFACE

The off-line user interface consists of a database support system called the off-line support system (OSS) which allows data management for the real-time simulation models. The PECS system uses the ORACLE™ relational database management system and other ORACLE™ products such as SQL*FORMS, SQL*MENU, and ProFortran. Designed using relational database techniques, the database user interface makes data available for update using menu driven interactive screen inputs and standard SQL commands. The database support system is divided into four major categories:

1) Library Maintenance
2) Mission Preparation
3) Configuration Management
4) Miscellaneous Function.

Library Maintenance involves editing threat parametrics, EW avionics data, and threat related information which is used to support a number of EC simulation functions.

The threat parametrics portion of library maintenance includes data used by the real-time threat and weapon models to perform an accurate simulation of the threats as they interact with the EC environment. The threat

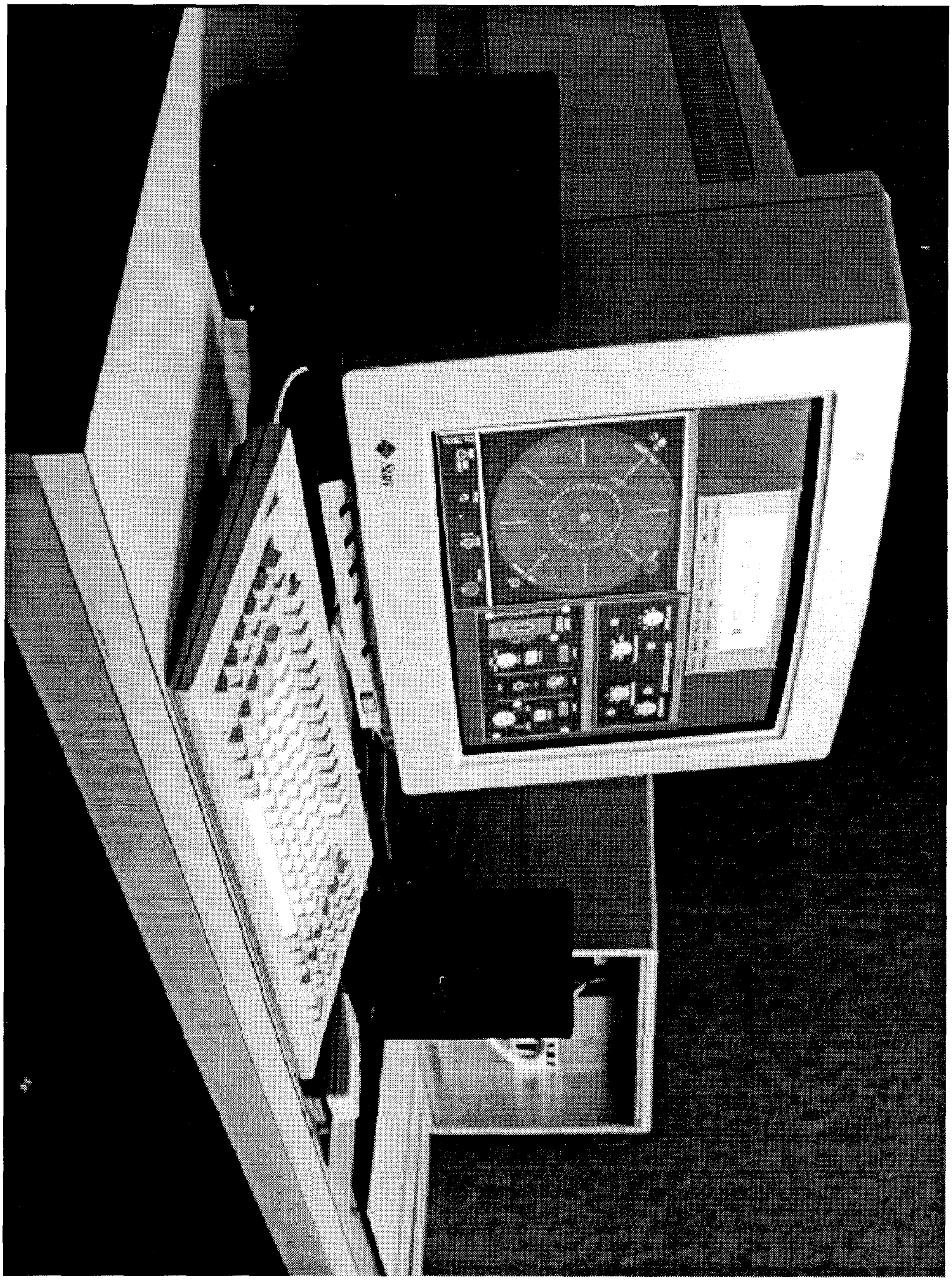


Figure 4. PECS System

parametrics include emitter parametrics and tactics data by mode and emitter, weapon data, and general threat information.

The aircraft EW configuration portion allows specific aircraft system setup data to be defined. This includes system warm-up and cool down times, the positioning of countermeasures dispense hardware onboard the aircraft, and equipment field of view limitations.

The EW system characteristics and EID libraries contain data used by the various EW system models to perform accurate simulations of the EW systems against threats in the environment.

The EOB/C3 library defines engagement modifiers, mode time modifiers, fire time delays, and communication delay times based on a particular conflict level. The conflict level defines the skill level or alert status of enemy forces.

The sites and platforms library maintains a list of threats associated with a site or platform entity. The site is a static entity which allows the user to construct real threat sites using JARM entities. The platform is a dynamic entity which allows the user to associate JARM entities to an object capable of being moved around in the real-time gaming scenario. The platform can be defined as either a computer controlled airborne interceptor (AI) or as a ship/AI platform. The computer controlled AI automatically moves and engages targets in the gaming scenario based upon positional information. The ship/AI platform allows the user to move the platform around the gaming scenario using the mouse or trackball. The sites and platforms built using the OSS can be positioned in a mission scenario causing all associated JARMs to be defined in the gaming environment.

The Mission Preparation function involves the building of mission scenarios, or threat laydowns for the real-time EC simulation. The mission preparation function allows the user to construct a mission load which contains up to 25 mission files consisting of up to 400 threats in each mission file. This mission load is downloaded to the real-time SCSI disk drive and is read during scenario initialization. The mission load also includes all threat and weapon parametrics, EW system characteristics and emitter identification data, and threat positional information required to support the real-time EC simulation.

The mission laydown is the EC gaming environment defined by the user to accomplish a specific part task training or mission rehearsal objective. The mission laydown is assembled by positioning threats using the latitude/longitude coordinate system. Each threat is assigned other pertinent information such as altitude, speed, heading, C3 data, and site/platform information. The mission laydown also defines which threats are networked together or are autonomous in the gaming environment.

The PECS system can be interfaced via an ethernet LAN with other mission planning systems so that mission laydowns can be transferred from the mission planning system to the OSS using standard file transfer protocols.

PERFORMANCE SUPPORT FUNCTIONS

The graphical user interface design for the performance support function (PSF) of the PECS system is an icon based, menu driven approach that allows easy access to the different task areas. For each task area, the PSF allows related information to be accessed. For example, with RWR operation, the user can obtain technical order information, view RWR operation in an automated fashion, or obtain a training lesson on RWR operation. Full audio narration is provided when requested for appropriate task areas and important warnings, cautions, and notes are amplified. Thus, the user can access related information in a myriad of different ways.

The PSF provides the capability of providing information concerning specific task areas related to EW avionics operation, checklist procedures, and recommended tactical operation. Currently, the system is limited to the EW systems currently used in the aircraft system previously mentioned. The function also provides a threat encyclopedia and a computer based training function.

The task area element is designed to provide refresher training and advanced concepts regarding EC to air crew members. The task area graphically details the operation and use of the selected EW avionics.

The threat encyclopedia function provides a graphical representation of the threat system with associated text and audio. The user can choose a selection from radars, missiles, guns, and jammers.

The CBT function provides a formal, step-by-step walk through of system operations, checklist procedures, and malfunction analysis.

SUMMARY

The PECS system provides a substantial EC training and mission rehearsal capability stuffed into a portable package. The system provides the flexibility and capability to train a wide variety of air crew member skill levels. And due to the use of COTS products, it can be more easily integrated with other training products (like portable weapon system trainers and mission planning systems) and its maintainability is greatly improved.

Although a great deal of development has occurred on the PECS system, a number of items can be added to improve training capability. For example, the use of digitized pictures with superimposed, animated knobs and buttons can be used for added realism. Digitized video can be utilized to better show EW system operation in the performance support functions to improve and maintain learning curves. 3D terrain images and cockpit displays can be added to expand training capabilities and realism. The PECS system COTS architecture allows for expandability, fidelity upgradability, and modularity.

The PECS system shows that effective EC part task training and mission rehearsal can be performed within a portable package using COTS products. Air crew member training and mission readiness is enhanced due to the fact that a system like this can be deployed quickly and in environments where larger systems cannot be employed. Maintainability is vastly improved with the use of COTS software and hardware products. Thus, overall system life cycle costs are substantially reduced while specific task training and mission readiness is improved.

REFERENCE

Galloway, D.W. , Heffernan, P.G., Nuss, E.A., Summers, C.M., Proceedings, 15th Interservice/Industry Training And Education Conference, Nov. 1993.

DISMOUNTED INFANTRY IN DISTRIBUTED INTERACTIVE SIMULATION

Robert W. Franceschini, Mikel D. Petty, and Douglas A. Reece
Institute for Simulation and Training
3280 Progress Drive, Orlando FL 32826-0544

ABSTRACT

Simulation of dismounted infantry in realistic numbers and behaviors was omitted from SIMNET, the prototype DIS-type simulation. Representation and simulation of dismounted infantry are not obviously fitted into the same framework as vehicles because models of humans are more complicated and not well understood. This paper describes a dismounted infantry simulation system developed at the Institute for Simulation and Training, and reports on lessons learned about simulating dismounted infantry in DIS-type simulations.

IST's Semi-Automated Forces Dismounted Infantry (SAFDI) project developed a Computer Generated Forces system with specialized capabilities for dismounted infantry. The goals of the SAFDI project are twofold: first, to provide a realistic simulation of dismounted infantry for the benefit of SIMNET trainees, and second, to learn about the simulation of dismounted infantry in support of future DIS simulations (like CCTT). The SAFDI system has been installed at training sites and has been used in training scenarios involving US Army soldiers. This paper provides an overview of the SAFDI system, including the project's goals, the system's capabilities, and the results of its evaluation at training sites.

IST's dismounted infantry research has led to a number of lessons learned of general applicability in the area of simulating dismounted infantry in DIS-type simulations, including SIMNET, BDS-D, and CCTT. This paper will address the following questions:

1. Why simulate dismounted infantry in DIS-type scenarios?
2. What are the distinctive characteristics of dismounted infantry that are important to its simulation?
3. How does one simulate dismounted infantry in DIS-type scenarios?
4. What mistakes were made in the design of SIMNET that made retrofitting it with dismounted infantry problematic?
5. How well does the emerging DIS network protocol standard support special requirements of dismounted infantry?

BIOGRAPHIES

Robert W. Franceschini is a Principal Investigator, Computer Generated Forces at the Institute for Simulation and Training. He has three years of experience on Computer Generated Forces projects at IST as a Research Assistant, Software Engineer, and Principal Investigator; he has seven CGF publications. Mr. Franceschini received a B.S. in Computer Science from the University of Central Florida. He is currently pursuing an M.S. in Computer Science from UCF.

Mikel D. Petty is Program Manager, Computer Generated Forces at the Institute for Simulation and Training. He is responsible for guiding Computer Generated Forces research projects at IST; presently there are five such projects. Mr. Petty received a M.S. in Computer Science from the University of Central Florida and a B.S. in Computer Science from the California State University, Sacramento. He is currently a Ph.D. student in Computer Science at UCF. His research interests are in simulation and artificial intelligence; he has over fifteen publications in those areas.

Douglas A. Reece is a Principal Investigator, Computer Generated Forces at the Institute for Simulation and Training. He received his B.S. and M.S. in Electrical Engineering from Case Western Reserve University and a Ph.D. in Computer Science from Carnegie Mellon University in 1992. He has been with IST since 1992. His research interests include autonomous agent design, computer vision, simulation, and traffic modeling.

DISMOUNTED INFANTRY IN DISTRIBUTED INTERACTIVE SIMULATION

Robert W. Franceschini, Mikel D. Petty, and Douglas A. Reece
Institute for Simulation and Training
3280 Progress Drive, Orlando FL 32826-0544

INTRODUCTION

Dismounted infantry (DI) plays a crucial role in battlefield exercises, and is an important requirement for any battlefield training system. According to [O'Byrne,1993], "...the presence of dismounted infantry [is] indispensable on the virtual battlefield." Dismounted infantry soldiers are difficult to detect in the battlefield (they are much smaller than vehicle platforms and therefore have a large number of choices for concealment). However, dismounted infantry are armed with powerful weapons (e.g., Dragon, RPG16, Stinger, SA-7 Grail) and thus are a dangerous threat for enemy vehicle platforms.

For all intents and purposes, SIMNET, the prototype distributed interactive simulation, did not include dismounted infantry. This provided an unrealistic training environment. Vehicle crews moved about the battlefield without concern for hidden and dangerously armed infantry. M2 Bradleys were always considered to be mounted in training exercises and the M2 crews could not learn about appropriate dismount procedures. Since SIMNET was not equipped with DI, it is also an incomplete tool for analysis; it does not allow new tactics or weapons to be tested against DI and does not allow infantry tactics or weapons to be tested at all.

CCTT and future simulations will require dismounted infantry. However, SIMNET provides no opportunity for learning about infantry simulation in distributed interactive simulation. This paper describes one system that retrofitted SIMNET with dismounted infantry and discusses some of the lessons learned about simulating dismounted infantry in distributed interactive simulation (DIS).

This paper is organized into three main sections. First, we will identify some of the characteristics of DI that make it difficult to simulate in a DIS-type simulation. The second section will present the IST Semi-Automated Forces Dismounted Infantry (SAFDI) system. The final section will discuss the SIMNET and the DIS network protocol standards relative to DI simulation.

SIMULATING DI IN DIS

Dismounted Infantry Characteristics

Distributed interactive simulation of dismounted infantry is a difficult problem. Historically, DIS-type simulations have been developed which are able to represent individual ground vehicle platforms with physical dynamics models and behavior models; these models are well described using a relatively small set of data points (location, orientation, turret heading, etc.). However, there are many characteristics particular to dismounted infantry that make their simulation problematic.

<u>Physical characteristics.</u> Because dismounted infantry are humans, their physical characteristics are not easily described with a small number of data points, DI are humans, they are not vehicle platforms. They have many articulated parts, rather than one or two common on ground vehicle platforms.

<u>Behaviors.</u> DI behaviors are more complicated than vehicle behaviors. They involve numerous fine details of posture changes, formation changes, and use of micro-terrain.

<u>Groups.</u> Dismounted soldiers typically work in groups ("fireteams"). The composition and mission of these groups can change dynamically. Some DI behaviors and capabilities apply to the groups, while others to the individual soldiers.

<u>Missions.</u> DI soldiers have different missions. For example, one squad of dismounted infantry soldiers could be assigned as forward observers and armed appropriately; another squad could be assigned in an anti-aircraft capacity and armed with Stinger missiles.

<u>Mounting and dismounting.</u> DI-to-vehicle platform interactions are similar to vehicle-to-vehicle interactions (sighting, target acquisition, firing, etc.), with one important addition: DI have the ability to mount and dismount vehicles. Mounting and dismounting are surprisingly complicated operations if implemented in full detail. Some details that can be included are the

number of soldiers that can fit into a particular vehicle (such as an M2 Bradley Infantry Fighting Vehicle), the amount of time it takes for the soldiers to enter or leave the vehicle, the locations for mounting and dismounting to take place relative to the vehicle, how to handle situations when more soldiers are trying to mount a vehicle than can fit, and how to handle situations when the vehicle absolutely must move during mounting or dismounting.

Weapons. DI soldiers can use many different weapons. These weapons may be transferred between soldiers and can be dropped and picked up later.

Types of DI Simulation Systems

A further complication of dismounted infantry simulation is the purpose that dismounted infantry are to serve in the exercise. There are two types of dismounted infantry simulations: *DI generators* and *DI trainers* [Petty, 1994].

DI generators provide dismounted infantry to the distributed interactive simulation for the benefit of other participants. For example, vehicle crew training requires large numbers of dismounted infantry soldiers that are capable of a relatively small number of behaviors (see discussion of the SAFDI system below for example behaviors). Typically, generated DI are controlled by operators who are part of the simulation, not by the trainees taking part in the simulation; a single operator often controls more than one DI entity. DI generators provide entities at the squad or fireteam level.

In contrast, DI trainers are intended to train humans in DI skills by involving them in the battlefield simulation. In this case, the fidelity of the simulation is very important for the simulation operator, since that person is being trained. Team leader training requires a much higher resolution model of physical and behavioral characteristics to be effective. In this case, DI entities represent individuals in the battlefield simulation.

At this stage of dismounted infantry simulation development, DI generators do not act as DI trainers and vice-versa. The reason for this lies in the tradeoff between the number of DI entities that can be supported and simulation fidelity; effective DI generators handle many more entities than DI trainers, but DI trainers have much better fidelity than DI generators. Therefore, the choice between these two simulation types drastically affects the architecture of the dismounted infantry simulation system.

THE SAFDI SYSTEM

Overview

In order to address the limitations of the original implementation of dismounted infantry in SIMNET and to learn about how to simulate dismounted infantry in DIS-type simulations, STRICOM asked the Institute for Simulation and Training (IST) to extend its Computer Generated Forces (CGF) Testbed to simulate dismounted infantry. The resulting extension and specialization of the Testbed is known as the Semi-Automated Forces Dismounted Infantry (SAFDI) system. More information about SAFDI can be found in [Franceschini,1994]. The SAFDI system is an example of a DI generator.

Like the CGF Testbed upon which it is based, the SAFDI system is composed of two types of components: Simulators and Operator Interfaces. A Simulator consists of a personal computer and IST-developed software that generates and simulates dismounted infantry fireteams. An Operator Interface is a separate PC and software system through which a human operator issues commands and instructions to the Simulator. The Simulator, in turn, is responsible for carrying out those commands within the simulated environment.

Project Goals

Initially the SAFDI project had two goals. First, SAFDI served as a demonstration and test application for the IST CGF Testbed. Second, the SAFDI project demonstrated the feasibility for adding a cost effective dismounted infantry CGF component to a DIS battlefield and to future networked simulations. It is important to note that the SAFDI system was developed in support of training vehicle crews in SIMNET; it was not designed to be a trainer for dismounted infantry.

The project's success at meeting its initial goals resulted in an expansion of the work and three new goals: first, to provide a stable CGF dismounted infantry system for evaluation at SIMNET/BDS-D sites; second, to develop dismounted infantry capabilities in a CGF system; third, to build up experience in using dismounted infantry in a DIS-type simulation. The last goal has grown in importance over time.

Basic Capabilities

In the battlefield, entities generated by the SAFDI system have a substantial set of basic capabilities. They can:

- Sight activity within Line of Sight
- Report sightings to an operator via the Operator Interface
- Kill enemy infantry and vehicles using small arms or missiles
- Mount and dismount vehicles
- Be seen according to visual range and posture
- Be killed by hostile direct and indirect fire
- Change visual appearance based on posture
- Change movement speed

In addition to fireteams, the SAFDI system can generate and control certain types of vehicles closely associated with infantry; specifically, the SAFDI system can generate Soviet-made BMP and US M2 Bradley Fighting Vehicles (BMPs/BFVs) and Soviet-made T-72 and US M1 tanks. SAFDI vehicles have capabilities similar to those of the SAFDI fireteams: they can move, detect enemy entities, report sightings, fire weapons, and be destroyed. In addition, SAFDI BMPs/BFVs can transport SAFDI fireteams. BMPs/BFVs may be used in support of SAFDI fireteams or independently to flesh out a detachment of simulators in an exercise.

Advanced Capabilities

System parameter files. There are several configuration files that affect the SAFDI system. These files contain information that controls the SAFDI simulation. They are useful for playing "what if" scenarios and for scaling the proficiency of SAFDI entities. The following is a partial list of the values that are controlled through the configuration files:

- Probability that a fired round will hit an entity
- Probability of a kill
- Amount of damage suffered by an entity when hit by a particular round
- Sighting distances
- DI mount/dismount time
- Physical specification for entities (maximum speed, maximum amount of ammunition and missiles, location of weapons, etc.)
- Missile dynamics (initial speed, acceleration, maximum flight speed, etc.)
- Parametric fireteam configuration information (see description of parametric fireteams below)

SAFDI configuration files only affect the abilities of SAFDI entities; no other entities are affected.

The configuration files are plain text files. They can be modified with any text editor. The file format is designed to be easy to modify by site support personnel. Parameter values are completely described in the documentation accompanying the SAFDI system.

Group commands. SAFDI entities on one workstation can be grouped together in a command hierarchy (see Figure 1). This allows the operator to command many SAFDI entities with a single order, which makes it easier to use the system. Some commands that may be issued to groups include giving permission to fire; posture, speed, and heading changes; mount and dismount; and routing. Giving an order to a group is similar to giving an order to each individual member of the group, one at a time.

A group can be included as a member of another group. One application of this feature is to group SAFDI entities into different platoons, and then put the platoon groups into a single company group; this way, either the entire company or any platoon will receive a command while only one operator command was issued.

The SAFDI operator can give orders to an individual SAFDI entity even if the entity is part of a group. For example, a group could be commanded to route to a location two kilometers away; one tank in the group could be told to halt while the rest of the group continues on the route.

Attach and follow. A SAFDI entity can be ordered to attach to any simulated entity; the attached to entity becomes the leader of the SAFDI entity. When a SAFDI entity is attached to a leader, it will follow the leader when the leader moves. This capability operates by matching speed and direction; it does not attempt to preserve formations. It is intended to allow SAFDI DI to follow manned simulators or SAFDI infantry fighting vehicles.

Attach and Follow is useful in conjunction with the Fire When capability (which automatically gives an entity permission to fire when a specified entity fires). The combination of these two features provides a means to give tactical information to SAFDI entities, as follows. SAFDI entities can be ordered to Attach and Follow and Fire When a human-controlled entity (for example, a manned M2 simulator). The human-controlled entity makes tactical decisions which are mimicked by the SAFDI entity automatically.

New infantry icons for CIGs. A significant part of the hardware of manned tank simulators and Stealth devices is the computer image generator (CIG) for displaying a three-dimensional out-the-window view of the battlefield. As a result of the first design for fireteams,

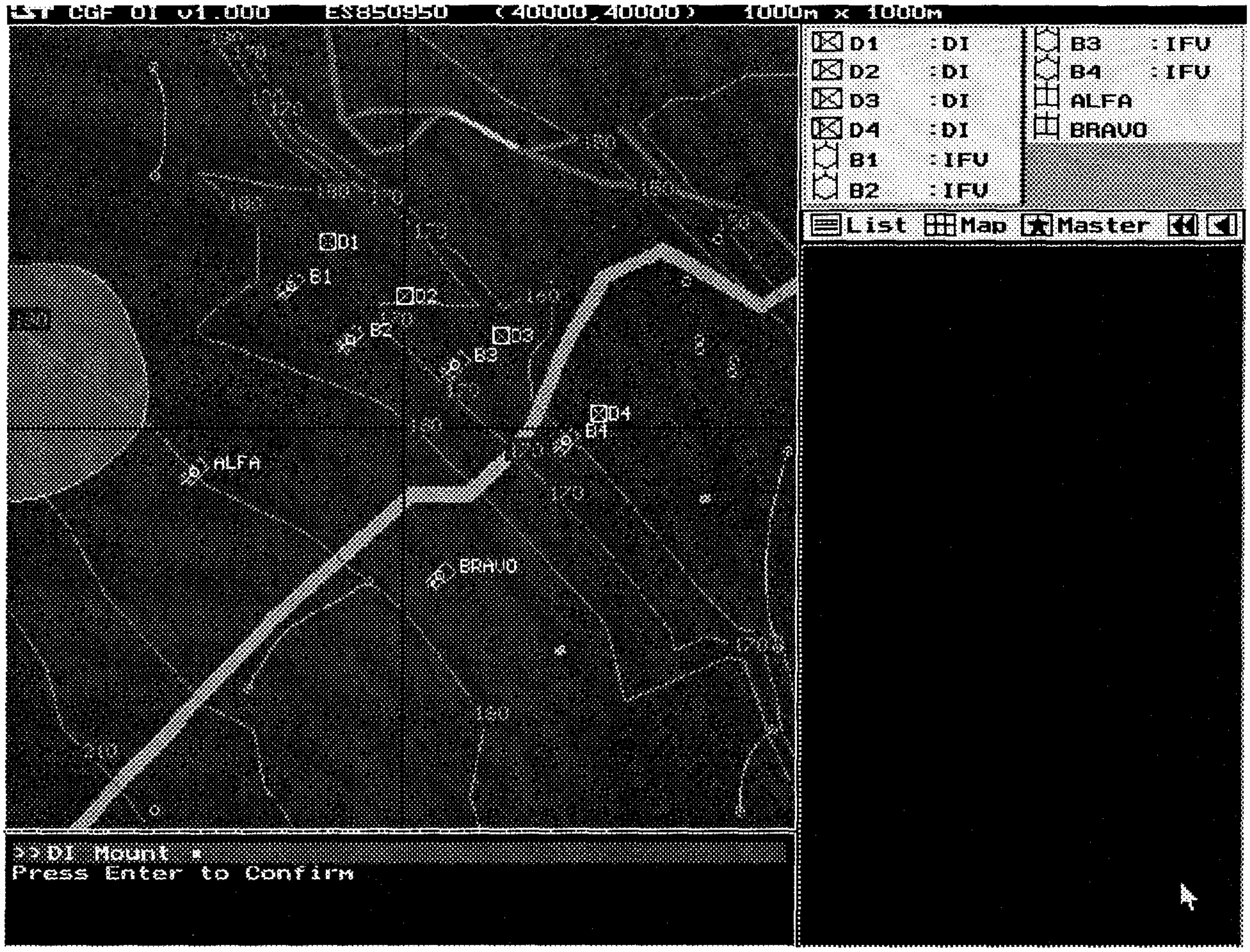


Figure 1 – Group Commands in SAFDI

the original SIMNET CIGs displayed one icon for each fireteam (group of five soldiers). This icon was composed of two perpendicular polygons with the picture of a soldier painted on each plane.

As part of the SAFDI development, IST created new visual icons for fireteams to be used in these devices. SAFDI's icons are compatible with SIMNET CIGs, and have been installed on the Stealth and manned M1 simulator at IST. These polygonal icons can be adapted to other CIGs. SAFDI will correctly use either the new IST developed icons or the standard icons; this can be controlled through the configuration files.

IST's new icons are three-dimensional human models with separate polygons for the head, arms, torso, legs, and weapons. The icons vary with alignment, posture changes, and weapon types. Some examples of the new icons include a sitting (kneeling) figure with a Dragon launcher, a standing figure with a shoulder SAM launcher, and a forward observer with field glasses.

In conjunction with the parametric fireteam capability (described below), IST developed the capability to use one icon to represent one soldier in the out-the-window CIG view. Each soldier is represented by an icon appropriate to the soldier's weapons, posture, and activity (e.g., whether the soldier is about to fire at a target). The SAFDI system automatically determines the appropriate icons to display for each soldier and sends the appropriate PDUs to other networked simulators.

<u>Air defense weapons.</u> SAFDI allows soldiers in a fireteam to use surface-to-air missiles. The missiles implemented are the Stinger and the SA-7 Grail. Air targets are prioritized based on whether their primary function is to engage targets from air to ground or from air to air. Within each of these divisions, rotary wing craft have higher priority than fixed wing craft.

Forward observers. SAFDI fireteams can be configured to include Forward Observers (FOs). FOs scan the terrain for enemy targets. When they acquire valid targets, FOs autonomously request indirect fire from the SAFDI system at the locations of the targets.

Parametric fireteams. Parametric fireteams allow specialized infantry to be used in battles. Rather than providing a fixed fireteam configuration, the SAFDI system allows the user to determine the make up of a fireteam in terms of the number of soldiers and the weapon types carried by each soldier.

In the SAFDI system, a DI entity represents a fireteam. Fireteams consist of one to six soldiers and are configured by the SAFDI operator or site personnel. Each soldier potentially carries an anti-tank missile (ATM), surface-to-air missile (SAM), squad automatic weapon (SAW), Grenade launcher, or rifle.

For example, a fireteam might consist of

3x Riflemen
1x SAW gunner
1x ATM gunner

Multiple fireteam compositions can be simultaneously defined and used by the operator. The definitions of fireteams are contained in the system parameter files (described earlier).

This representation of fireteams allows certain behaviors to be associated with the aggregate fireteam while other behaviors are associated with the individual soldiers. This allows computationally intensive operations (e.g., intervisibility calculations or routing) to be performed on the fireteam as a whole, but still allows certain operations to be performed for individuals (e.g., posture settings or firing weapons).

Soldiers within SAFDI fireteams are assessed damage as individuals rather than as an aggregate. Specific damage and suppression results are probabilistically calculated for each member of the fireteam when attacked. These results are associated with the particular soldier and remembered for future attacks.

In SAFDI, there is currently no simulation of wounding individuals; individuals are either alive or killed.

Delivery and evaluation. IST delivered the SAFDI system to SIMNET sites in two phases: the Basic and Enhanced SAFDI systems. The Basic SAFDI system was delivered during September 1993 while the Enhanced SAFDI system was delivered during February 1994. At each delivery, IST installed the SAFDI system on the site's computers, gave a capabilities demonstration, and conducted operator and support personnel training.

After delivery, the SAFDI system was evaluated at the Ft. Benning training site by a team of site personnel and simulation experts. The evaluation consisted of a series of training scenarios where A Company 1/29 Infantry fought with and against dismounted infantry generated by the SAFDI system. During the evaluation, the SAFDI system was operated by the evaluators, rather than IST personnel.

The results of the evaluation are discussed in [D'Errico,1994]. In general, the evaluators found many opportunities for improvement in the SAFDI system, but overall were very pleased with the system's performance and the realism added to the simulated battlefield. Captain William Hessenius, commander of A Company, said, "SAFDI greatly increased my unit's training" [D'Errico,1994].

Here, we summarize some of IST's observations from the evaluation of the Basic SAFDI system (IST has not observed the evaluation of the Enhanced SAFDI system).

SAFDI was completely reliable throughout the evaluation; it ran for a total of approximately 16 hours in three training scenarios without crashing even once. This is especially impressive considering that one of the scenarios contained over 120 entities (which is three times SAFDI's rated capacity). In comparison, the production site semi-automated forces system crashed three times during evaluations.

When dismounted infantry were added to the simulation, the pace of battle slowed dramatically. In one instance, Fort Benning's site personnel estimated that one of the scenarios would have taken about 35 minutes to complete without dismounted elements; that scenario took two hours with DI.

SAFDI entities participated realistically in the battles. They engaged enemy targets, mounted and dismounted friendly infantry fighting vehicles, etc.

BATTLEFIELD NETWORK PROTOCOLS AND DISMOUNTED INFANTRY

DI in SIMNET

The original SIMNET implementation had some idiosyncrasies that made retrofitting it with dismounted infantry difficult. These SIMNET problems are listed in order to avoid making the same mistakes in future network protocols like DIS.

One icon per fireteam. In SIMNET, the standard dismounted infantry entity represents a fireteam. This fireteam consists of five soldiers, but is rendered by the SIMNET image generators as a single icon depicting a single soldier. Therefore, there is no visual cue to a battlefield observer to indicate the number of soldiers in the fireteam that are capable of fighting. In the evaluation of the Basic SAFDI system, the soldiers identified this as the worst feature of the simulation (note that this was an inherent problem in SIMNET; as mentioned earlier, IST has developed an experimental solution to this problem) [D'Errico,1994].

No mount or dismount procedures. Although SIMNET is equipped with a Dismounted Infantry Module (DIM) that can mount and dismount vehicles, there is no formal exchange of information on the network when a mount or dismount takes place; mounting and dismounting are not included in the SIMNET protocol. Therefore, vehicle crews have no easy way of knowing when mounting or dismounting is taking place (in an early version of SAFDI, the dismounted infantry fireteam was positioned in front of the manned simulator during mounting so that it could be seen through the vision blocks of the vehicle crew). Furthermore, there is no protocol in SIMNET for determining whether a vehicle is mounted.

No coaxial machine guns on simulators. Coaxial machine guns are the weapon of choice for ground vehicle platforms against dismounted infantry. However, standard SIMNET manned simulators (e.g., M1 and M2) are not equipped with coaxial machine guns. Therefore, dismounted infantry can be engaged in one of three ways in SIMNET by vehicle crews: by radioing for artillery support, by colliding with the dismounted infantry, or by using a non-standard weapon (such as the main gun) against a fireteam. While the first two cases are acceptable in terms of battlefield realism, the third case is not; one could argue that this situation has a negative training effect.

DI in DIS

While the existing DIS standard has a mechanism for representing small dismounted infantry units, this mechanism is crude and is really only an afterthought to the primary goal of representing tanks and other vehicles. This section discusses several specific limitations of DIS for DI: the lack detailed entity state information for life forms; the limited representation of objects; the lack of detail in small arms firing events; and the lack of dynamic terrain.

Life form state representation. The DIS specification (version 2.0.4 [IST,1994a]) attempts to classify many types of life form entities, but the classifications in the entity type record are inappropriate. The first problem is that aspects of entity *activity* or *appearance* are used as *type* characteristics. For example, parachutists, dismounted infantry, and swimmers are separate types. The same entity could take on all three roles during an exercise, requiring that its type change. These particular actions are in fact duplicated in the life form appearance record! The second problem is that the type record determines the type of the entity from the weapon it is carrying. Entity types should not be tied to such objects, which the entity may drop, pick up, or exchange in the scenario. Furthermore, most DI entities will carry more than one weapon; the entity type is then ambiguous.

The DIS appearance record for life forms allows for prone, kneeling, and upright states; and for crawling, walking, running, and jumping "gaits." This limited set of states is not adequate for simulations that allow close inspection of the entity, such as individual-level simulations. In an individual-level simulation, many other states are also desirable—leaning around a corner, crouching below cover, twisting the body or head to look or direct a weapon in a different direction than the feet are pointing, etc. Simulations will also soon require explicit communication between soldiers; at the individual level, this is often done with arm signals. Thus detailed arm positions may have to be included in the soldier representation.

The need for detailed state representation for body position leads to questions about how the overall DIS design will scale up when used for individual combatants or other high-resolution DI exercises. The detailed representation requires a much higher bandwidth than is usual for vehicles or aggregate entities. Not only does this requirement load the network, but all other entities must process the high-resolution state information. The bandwidth requirement could possibly be reduced by using "intelligent" dead-reckoning algorithms that can generate detailed body movement from abstract "task" descriptions; soldier tasks would presumably change far less often than joint angles. Furthermore, entities that don't need detailed state information can omit the reconstruction of detailed posture. However, this approach has the problem that the dead-reckoned reconstruction of entity movements may not correlate well enough with the actual entity movements. The dead-reckoned movements may also lag the actual movements, since the task abstraction in effect low-pass filters the entity movements. These errors could be crucial when sighting and firing opportunities pass quickly.

Object representation. The representation of objects in DIS is essentially limited to entities. Having representations for other objects would be useful in general, but is especially needed for DI. Fireteams (and even individual soldiers) typically carry many pieces of equipment and weapons that they use in combat. They carry ammunition for themselves and squad weapons, pieces of squad weapons, grenades, etc. These objects are held in different positions, used, stowed, expended, dropped, picked up, put in other objects, and given to other soldiers. The Destructible Entity protocol in DIS version 3.0 [IST,1994b] would be awkward to use for objects because they may change state frequently. A new component must be added to DIS to represent such objects.

Weapons fire. Weapons fire is represented in DIS with a Fire PDU followed by a Detonation PDU. Each of these can indicate in a Burst Descriptor that the event contained multiple rounds. However, there is no provision to indicate where each round went. This limitation is not acceptable in a detailed simulation with DI because each round may be significant (especially at the individual level). In addition to normal scatter from weapon movement, rounds may ricochet and cause multiple impacts. At the very least, the Fire PDU should indicate a standard scatter pattern.

In addition to impact locations, DI simulations also need to compute munition trajectories (at least roughly) because soldiers hear the rounds passing by even if they don't impact nearby. The trajectories cannot always be computed from impact locations because rounds may leave the terrain database before impacting. Fire PDUs must therefore provide basic trajectory information.

Finally, the life form appearance record allows for weapons to be stowed, deployed, and in the firing position. However, there are no definitions associated with these states, and at any rate three states may not be sufficient. A rifle, for example, could be slung on the soldier's back in one of several ways, held in one hand several ways, held with both hands several ways (e.g., down, level at waist, at chest level, across the arms, at eye level, pointing up, etc.), held in a sling, etc. These postures may be important for determining a soldier's status, activity, or intentions.

Dynamic terrain. As with objects, dynamic terrain will eventually be useful for all domains of DIS simulation. Dynamic terrain will be crucial to detailed DI simulations because the soldiers interact with the environment to such a great degree. In fact, at the individual level the distinction between objects (which presumably can be moved during a simulation) and terrain features is fuzzy. Rocks, trees, debris, doors, windows, etc. are part of the terrain, but can all be moved, changed, avoided, jumped, used for cover, etc. by soldiers. The current DIS design does not have an adequate mechanism for representing dynamic terrain.

The solutions to most of these problems seem to require increased computation and increased network bandwidth. We are attempting to develop solutions that will allow the DIS simulators to provide an adequate environment for training while limiting the increase in performance required in the system.

CONCLUSIONS

Dismounted infantry soldiers are more difficult to represent in distributed interactive simulations than most of the vehicle platforms typically involved in DIS-type exercises. However, they provide a key component which dramatically affects tactics in the battlefield. Therefore, their inclusion in distributed interactive simulations is crucial to successful future training and analysis.

This paper has outlined the issues involved in representing dismounted infantry in distributed interactive simulations. Dismounted infantry pose a technical challenge to DIS-type exercises for two reasons: human models are inherently complicated and the simulation community continues to think of new ways that human models should be used. Models of dismounted infantry that are used as the basis for future network protocols must include some method of being simple but extensible.

REFERENCES

D'Errico, J. (1994). "Evaluation of the SAFDI System at the USAIS, Ft. Benning, GA SIMNET Site", Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation, Institute for Simulation and Training, May 4-6 1994. pp. 149-154.

Franceschini, R.W. and Petty, M.D. (1994). "Dismounted Infantry in DIS-type Scenarios: A SAFDI Project Overview", Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation, Institute for Simulation and Training, May 4-6 1994, pp. 155-167.

Institute for Simulation and Training, (1994a). "Enumeration and Bit-encoded Values for use with IEEE 1278.1-1994, Distributed Interactive Simulation -- Application Protocols", Technical Report IST-CR-93-46, Institute for Simulation and Training, March 1994.

Institute for Simulation and Training, (1994b). "Standard for Distributed Interactive Simulation -- Application Protocols, Version 3.0 Working Draft", Technical Report IST-CR-94-18, Institute for Simulation and Training, 7 March 1994.

O'Byrne, E.C. (1993). "Dismounted Infantry: Indispensable to the Virtual Battlefield", Proceedings of the 15th Interservice/Industry Training Systems and Education Conference, Orlando FL, November 29-December 2 1993, pp. 783-791.

Petty, M.D. (1994). "Dismounted Infantry in Virtual Battlefield Simulation", Proceedings of the Individual Combatant Modeling and Simulation Symposium (INCOMSS-94), Fort Benning GA, 15-17 February 1994, pp. 288-305.

ACKNOWLEDGMENT

This research was sponsored by the US Army Simulation, Training, and Instrumentation Command as part of the Semi-Automated Forces Dismounted Infantry project, contract N61339-93-C-0026. This work built upon previous results from the Intelligent Simulated Forces project, contract N61339-89-C-0044, also sponsored by STRICOM. That support is gratefully acknowledged.

HIGH FIDELITY VIRTUAL PROTOTYPING TO SUPPORT GROUND VEHICLE ACQUISITION

Jon G. Kuhl, Ph.D.
Center for Computer-Aided Design, The University of Iowa
Iowa City, IA

LTC James Wargo, Ph.D., P.E.
Advanced Research Projects Agency
Arlington, VA

ABSTRACT

This paper describes an ARPA initiative to develop a comprehensive simulation-based design environment for ground vehicles. A central component of this environment is the use of high fidelity, operator-in-the-loop simulation for virtual prototyping, a necessity if the user is to participate actively and meaningfully in the development of a new ground vehicle.

A ground vehicle virtual prototyping capability is being developed, using the Iowa Driving Simulator (IDS) that employs real-time vehicle performance models with engineering detail comparable to models typically used for off-line design and analysis purposes, and employs terrain models that characterize surface type and geometry at fine resolution. This fidelity allows factors that previously could be evaluated only via physical prototypes to be evaluated through virtual prototyping, including detailed operator-vehicle performance characteristics and collection of specific vehicle performance data, such as component load histories, in realistic operational scenarios.

A "virtual proving ground" demonstration project conducted in July 1994, is described. For this test, the environment of two Aberdeen Proving Ground test courses has been duplicated on the IDS. A series of instrumented tests were conducted on the actual Aberdeen course and in the IDS-based virtual prototyping environment. Data, ranging from basic human factors measures to specific vehicle performance parameters, was collected and compared to assess the ability of the virtual environment to represent real-world conditions.

The paper also discusses additional aspects of the ARPA project, including ties to the synthetic battlefield, development of reconfigurable, virtual-prototyping environments, and integration of the virtual prototyping capabilities into a comprehensive integrated product and process development (IPPD) framework.

ABOUT THE AUTHORS

Jon G. Kuhl, Ph.D., is a Professor of Electrical and Computer Engineering and Deputy Director of the Center for Computer Aided Design at The University of Iowa. He directs the Center's vehicle virtual prototyping research program and is a co-principal investigator on the US DOT-sponsored National Advanced Driving Simulator project.

James Wargo, LTC US Army P.E., received his Ph.D. from the University of Texas (Austin). He is currently a Program Manager in the Advanced Systems Technology Office at ARPA. He previously managed the transition of SIMNET from ARPA to the Army.

HIGH FIDELITY VIRTUAL PROTOTYPING TO SUPPORT GROUND VEHICLE ACQUISITION

Jon G. Kuhl, Ph.D.
Center for Computer-Aided Design, The University of Iowa
Iowa City, IA

LTC James Wargo, Ph.D., P.E.
Advanced Research Projects Agency
Arlington, VA

INTRODUCTION

As the nature of world conflict shifts from superpower confrontations to lower intensity regional disputes, new demands, both tactical and fiscal, are being placed upon military battlefield systems. Future deployable, manned systems will need to focus on a number of characteristics, including reduced weight, smaller crew size, increased mobility, and enhanced reliability/maintainability. The technical challenges involved in meeting these objectives are exacerbated by increasingly tight resource constraints.

The ARPA Advanced Systems Technology Office (ASTO) has initiated a program to develop and demonstrate a simulation-based Integrated Product and Process Development (IPPD) framework to support the acquisition of future military ground vehicle systems. This environment will support rapid system redesign in response to changing requirements, and will facilitate simultaneous consideration of performance, operation, and support objectives throughout the development process. A cornerstone of this environment is the extensive use of advanced simulation at all stages in the design process. One important element of the simulation-based design environment is the use of soldier-in-the-loop virtual prototyping of design alternatives. This unique capability will allow the performance ramifications of design decisions to rapidly and cost-effectively assessed prior to physical prototyping and will provide a qualitatively new capability to tune system performance to the abilities of the human operator.

As an initial step in the establishment of this capability, a project is being conducted at The University of Iowa's Center for Computer-Aided Design, to demonstrate the feasibility of constructing and utilizing a "virtual proving ground" environment to support ground vehicle design and test objectives. This project is being directly assisted by the Combat System Test Activity (CSTA) and the Army Test and Evaluation Command (TECOM), with additional cooperation and participation by other Army Commands, including TARDEC, ARL, AMC/IOC, AMSAA, WES, and STRICOM. Ultimately, this virtual proving ground will provide an integrated framework for test and evaluation (T&E) at all phases in the design process. Initial efforts are aimed at demonstrating a simulation-based environment that replicates and extends the capabilities of the physical proving grounds currently utilized for ground vehicle T&E. The specific objectives are to produce a high fidelity, operator-in-the-loop simulation environment that closely replicates off-road test courses at the CSTA's Aberdeen Proving Grounds (APG) and to demonstrate the ability to effectively utilize this environment for an engineering design trade-off analysis. In the longer term, this environment will be fully integrated with the overall IPPD framework, allowing evolving designs to be tested on the virtual proving ground well in advance of the existence of hardware prototypes. The high fidelity ground-vehicle virtual prototyping capability will also be extended to the synthetic battlefield, via a Distributed Interactive Simulation (DIS) link, further broadening the realm of T&E to include representative tactical environments. The integration of such a high-fidelity engineering node with the synthetic battlefield will permit comprehensive assessments of new weapons systems concepts with the user in the loop at all stages of the development--an aspect that is missing in many concepts for virtual prototyping.

VIRTUAL PROTOTYPING ON THE IOWA DRIVING SIMULATOR

The University of Iowa Center for Computer Aided Design (CCAD) has an extensive research and development program in ground vehicle simulation and virtual prototyping. Supporting this program is a suite of simulator facilities, including the Iowa Driving Simulator (IDS), which is a state-of-the-art operator-in-the-loop ground vehicle driving simulator. The IDS is a highly reconfigurable facility that can simulate a variety of vehicle types and configurations, terrain conditions, and operational scenarios. The simulator employs high fidelity visual, motion, audio, and control force feedback and uses detailed computational vehicle modeling approach to represent the performance of the subject vehicle at an engineering level of fidelity. A first principles-based modeling approach is utilized so that real-time computational models of new or conceptual vehicle designs can be developed directly from engineering CAD data. A unique aspect of the IDS is its ability to represent terrain surface characteristics and associated vehicle-terrain interactions, both on-road and off-road, at an engineering level of resolution.

The IDS consists of a projection dome, mounted on top of a hexapod motion platform with 60-inch stroke actuators, providing full 6 degrees of freedom motion cueing (see Figure 1). Interchangeable vehicle cabs are mounted in the dome. Four channels of high resolution, textured graphics are projected on the inner dome surface, providing a nominal forward field of view of approximately 200 degrees and a 60 degree rear field of view. Viewing areas are highly reconfigurable to support different vehicle configurations and experimental requirements. Directional audio cueing is provided by a 16 voice digital sampling system, coupled to a MIDI mixer, driving 8 speakers positioned around the vehicle cab. realistic force feedback is provided to vehicle controls. Where appropriate, control loading forces are derived directly from the underlying computational vehicle dynamics model.

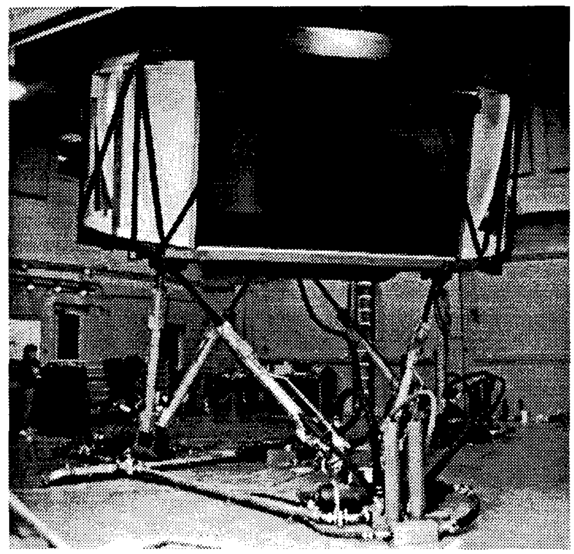


Figure 1. Iowa Driving Simulator

Two features of the IDS are critical to its suitability for high fidelity virtual prototyping applications. These are i) the high fidelity modeling approach used to simulate subject vehicle performance, and ii) the high resolution terrain surface characterization approach. The IDS vehicle dynamics subsystem computes subject vehicle performance based upon solution of differential-algebraic equations of motion derived from a first-principles multibody dynamics model of the vehicle. This modeling approach directly accounts for the kinematic and dynamic properties of the vehicle chassis and suspension. Since the modeling is based upon first principles--i.e. the dynamics formulation is derived directly from the kinematic joint-body structure of the mechanism--it is possible to accurately model new vehicles or proposed modifications in existing vehicles prior to existence of a physical prototype. This is important for virtual prototyping applications. The vehicle performance models traditionally utilized in operator-in-the-loop simulators are linearized, lumped-parameter models that must be imbued with empirically derived performance parameters to accurately characterize the behavior of a given vehicle system.

A typical IDS vehicle model for virtual prototyping purposes is shown in schematic form (see Figure 2). The level of fidelity, i.e., the number of degrees of freedom, in this model far surpasses that which is typically used in simulations for training and operations. Such detail is essential to support engineering trade-off analyses among design alternatives. This model is for an Army HMMWV. The model consists of 14 rigid bodies connected by various types of joints or other kinematic constraints. The letters that label joints in the figure denote: spherical joint (S), translational joint (T), revolute joint, and distance constraint (D). The bodies represent the chassis, steering rack, double A-arm suspension components and wheel assemblies. The geometry and mass properties of each body are captured by the model. For simplicity, the steering system has been approximated in this model by a simple translational joint that models a rack-and-pinion steering system. The steering tie rods are modeled as distance constraints. The resulting closed-loop kinematic structure of the vehicle is shown (see Figure 3). The model is augmented with appropriate force elements to represent springs, shock absorbers, etc.

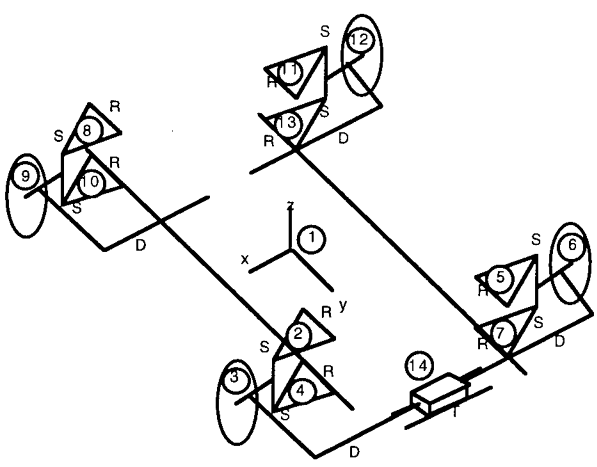


Figure 2. Schematic of HMMWV Vehicle

The nonlinear equations of motion governing the dynamic behavior of the model are formed

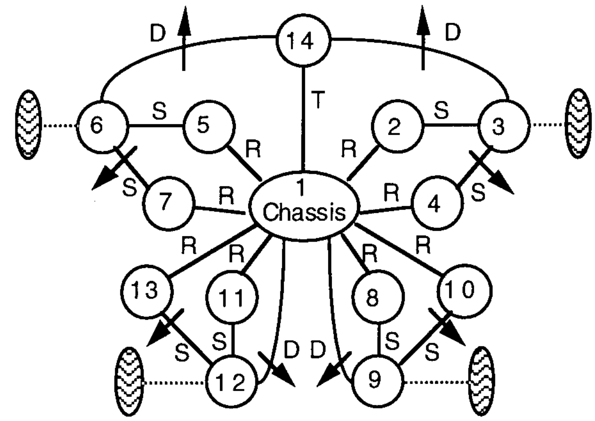


Figure 3. Topographical Graph for HMMWV 14-Body Model

and solved in real-time on an 8-processor parallel processing system using a real-time recursive dynamics (RDRD) package developed at The University of Iowa [1-3]. Current computer system performance allows the RTRD to achieve real-time performance using fixed integration time steps of approximately 200 Hz. This is sufficient to capture most frequencies of interest in the vehicle suspension. Where necessary, faster integration rates can be used for mathematically stiff vehicle subsystems, such as tires, by employing multi-rate integration schemes.

The basic dynamics model (see Figures 2 and 3) is augmented with models of various subsystems that act upon, or are acted upon by, the dynamics model, to form a complete vehicle model. The complete vehicle model is illustrated (see Figure 4). The complexity of each subsystem model is determined by the objectives of the application. For virtual prototyping applications involving wheeled vehicles, tire models are of critical importance Tire modeling approaches are discussed in detail in [4,5] A typical tire model computes tire forces and torques at each integration time step and transfers them to the wheel body within the multibody dynamics model.

Since a ground vehicle operates directly upon the terrain surface, and since interactions between the vehicle's tires (or tracks) and the terrain surface are of first-order importance to vehicle performance, accurate

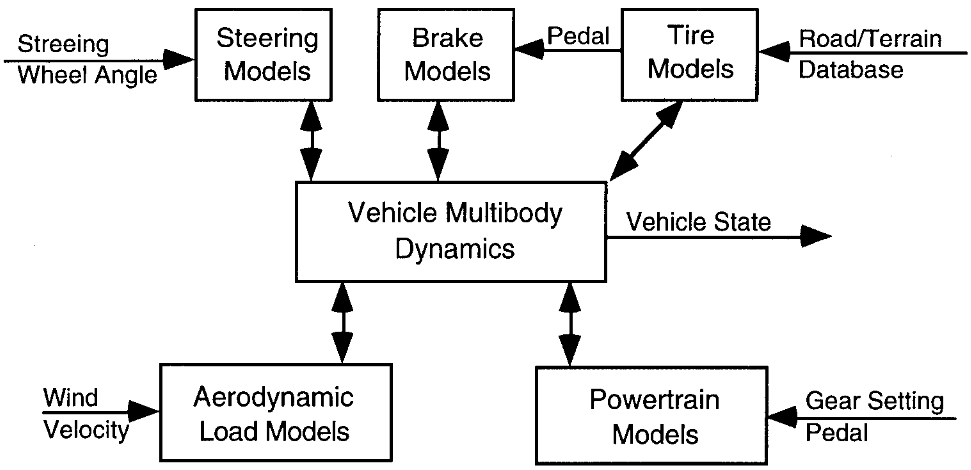


Figure 4. Vehicle Subsystem Modeling

representation of the terrain surface, in terms of both surface geometry and type, is essential for high fidelity operator-in-the-loop virtual prototyping. The IDS employs a high resolution terrain database that can represent surface geometry at arbitrarily fine levels of resolution. This database is automatically generated simultaneously with the creation of the visual database model for the computer image generator to insure correlation. Typically, the terrain model must contain much higher resolution than the polygonal representation used by the visual database model. In current applications, resolutions down to a few inches are routinely used to capture detailed surface characteristics. The database employs variable resolution, so extremely high resolution may be used only where it is actually needed. In addition to surface geometry, the terrain database maintains full surface type information. This information can include the composition of the surface (concrete, asphalt, gravel, dirt, etc.) and any other relevant data. An example of high resolution terrain modeling is given (see Figure 5). This shows a small section of the terrain model for a sinusoidal bump course. The terrain grid resolution is approximately 6 inches to accurately capture course geometry. Again, the engineering level of detail exceeds that associated with training and operations models because it is necessary for design T&E purposes, and to input sufficient detail to the high-fidelity systems models previously described. The terrain modeling methodology and correlated database generation approach is described in more detail in [6,7].

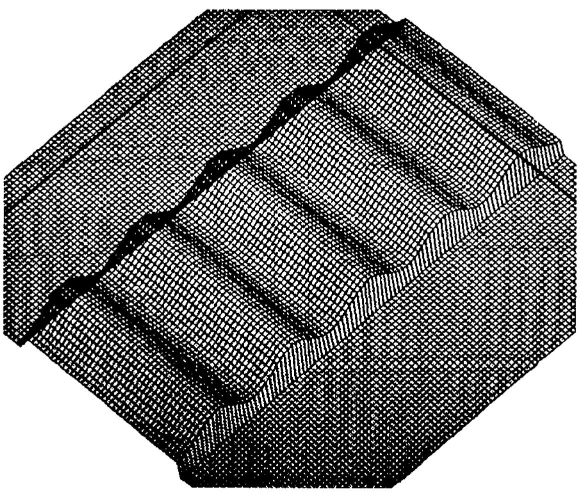


Figure 5. Portion of APG Terrain Grid

THE VIRTUAL PROVING GROUND DEMONSTRATION PROJECT

As part of the ARPA-sponsored activity described earlier, a demonstration project was conducted in July of 1994, to assess the feasibility of creating a simulation-based proving ground environment for ground vehicle T&E. The intent of this project was to reproduce portions of the Army TECOM'S Aberdeen Proving Ground facility in the simulation environment of the IDS, and to benchmark the capability to conduct useful engineering T&E in the resulting virtual proving ground environment. The Combat System Test Authority (CSTA) actively participated in this demonstration.

Two test courses on APG, Munson and Churchville, were modeled for the Iowa Driving Simulator. The modeling process included development of visual database models that closely represent the courses, both with respect to terrain geometry and cultural features (trees, buildings, etc.), and associated high-resolution, correlated terrain databases that model the terrain surface as closely as possible, based upon available measurements and surveys. The Munson Test Course, the smaller of the two courses, consists of a two mile closed dirt track with moderate terrain elevation changes, along with an 800 foot long sinusoidal washboard course and a skid pad. The Churchville test course is a four mile closed-track cross-country course with extremely steep grades (up to 29%), many hairpin turns, and considerable road roughness, including berms and ditches. A picture of the basic terrain geometry for the Churchville course is shown (see Figure 6). The basic terrain skin for this course was produced from a special, five-meter resolution, digital terrain elevation map produced by the Army Topographic Engineering Center. Additional course detail was derived from surveying data and engineering drawings.

For the July demonstration, experienced test drivers piloted an instrumented M1025 HMMWV over the APG courses. Approximately 40 channels of vehicle and human performance data will be collected during these runs. The same drivers then completed an identical scenario in the virtual proving ground environment of the IDS. Performance variables from the test track and simulator environments are currently being correlated to assess the degree of real-world engineering and human performance fidelity that has been captured in the virtual proving ground environment. The resulting comparison between the simulation-generated results and the physical measurements taken on the proving ground courses will indicate how far current technology must be advanced to allow operator-in-the-loop virtual prototyping to play a central role in the ground vehicle T&E process.

THE ROLE OF THE VIRTUAL PROVING GROUND IN IPPD

The joint ARPA-TECOM-Iowa project described above is not intended to demonstrate that simulation can obviate the

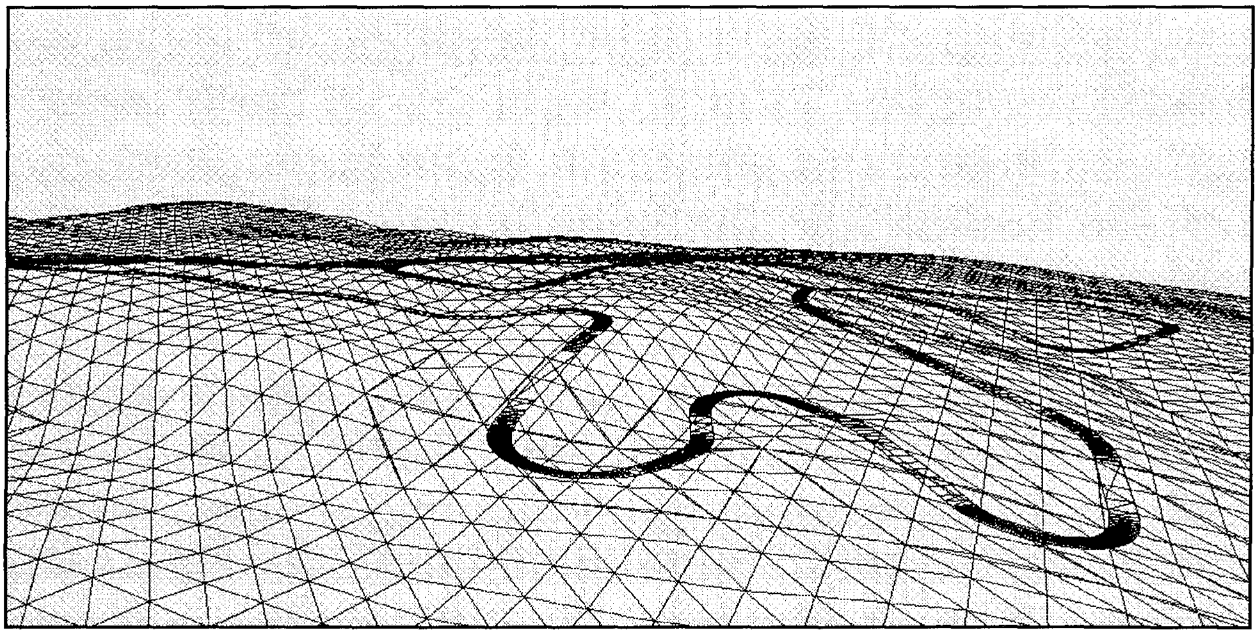


Figure 6. Complex Test Course Geometry

need for physical T&E of real hardware. Rather, it is intended to demonstrate that high-fidelity, operator-in-the-loop simulation can be exploited early in the design process for developmental testing. The likelihood for reduced hardware T&E is high, but more important is the potential for more robust T&E. By exploiting the virtual proving ground, vehicle program managers can show up at the real proving ground "smarter" with a more capable product. Likewise, by learning lessons on the virtual proving ground, testers can conduct "smarter," more focused testing. Ultimately the user operates a new system which has been far more thoroughly shaken out than it would have been by virtual or hardware testing alone. The long-term objective of the ARPA IPPD Simulation Program is to fully integrate the virtual proving ground into a comprehensive, integrated product and process development (IPPD) framework, to support the acquisition process for military ground vehicle systems. This framework is intended to integrate all phases of the acquisition process, including design, producibility, support, and cost.

The virtual prototyping capabilities embodied in the virtual proving ground will permit full scale soldier-in-the-loop T&E to begin well in advance of the existence of physical prototypes. This has the potential to significantly shorten the T&E process and save considerable physical prototyping costs, and allow more design iterations to be completed or design alternatives to be evaluated. In addition, it will provide a unique capability to address human-centered design issues throughout the development process.

The potential role of virtual prototyping in the military ground vehicle acquisition process is illustrated (see Figure 7). The figure highlights three significant contributions. The first of these is the potential for augmentation of traditional off-line engineering analysis functions as a result of realistic performance data derived from the virtual proving ground. For example, the assessment of component durability/ reliability requires load history and duty cycle data that is, in turn, determined by the manner in which the vehicle or weapon system is operated. Consequently, reliability and durability prediction tools have traditionally had to rely on assumed or estimated load data. The virtual proving ground provides a means of collecting more accurate engineering data to serve such analysis functions.

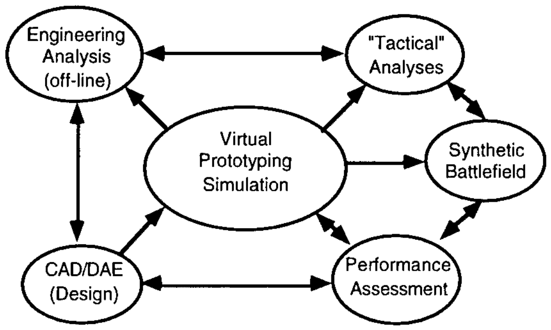


Figure 7. Role of Virtual Prototyping in the Acquisition Process

The second potential contribution of virtual prototyping to the acquisition process is the ability to address human-centered performance issues early in the development cycle. Currently, it is possible to place a human operator, or crew, in control of a new vehicle or weapon system until a full-scale physical prototype has been produced. By this time, many substantive design and manufacturing decisions have been locked in. By allowing human operation of new designs, or proposed redesigns, at earlier points in the design process, it will be possible to optimize human-machine performance much more effectively.

The third, longer term, role of virtual prototyping (see Figure 6) is the extension of engineering T&E into the tactical environment, represented by the synthetic battlefield. The ability to evaluate the performance of new ground vehicle systems in realistic tactical environments will provide more realistic data to support the development process a qualitatively new capability for design tradeoff analysis. It must be emphasized that, at the present time, a significant gap still exists between the engineering design environment and the synthetic battlefield environment in terms of overall fidelity and validity. The synthetic battlefield environment implemented by current distributed interactive simulation (DIS) architectures has been developed primarily to meet training and doctrine development objectives. Many issues must be resolved before a truly seamless integration of high-fidelity virtual prototyping with the synthetic battlefield can be achieved.

To support initial efforts toward this integration, the virtual prototyping environment of the IDS is currently being connected to the Defense Simulation Internet (DSI) as part of the BDS-D Version 1.0 System Test Bed. The mission of the BDS Test Bed is to demonstrate the interoperation of a highly heterogeneous, geographically separated network of simulation assets [8]. Several of the military and commercial facilities connected to the Test Bed are acquisition-oriented, including facilities of the Army Tank Automotive Command in Warren Michigan and engineering simulators for the Comanche and Apache Longbow Helicopter programs. As a result, the Test Bed will provide a suitable environment for investigation of DIS acquisition support issues.

SUMMARY AND CONCLUSIONS

Distributed simulation for training and operations has long been a high priority research area within the Department of Defense. For almost as long, the training community has pushed for similar emphasis by the acquisition community. In response to the challenge, researchers and developers have independently grown modeling and simulation capabilities with the goal of integrating them with the synthetic battlefield. The engineering-level of fidelity driving simulation described in this paper is a powerful addition to the DIS network. It bridges the gap between the synthetic battlefield which is so uniquely appropriate for addressing issues related to tactics and doctrine and the engineering world which is essential for design and development. The vision of virtual prototyping is to advance new concepts via simulation technology, simultaneously examining the added worth of the idea on the battlefield. This work furthers the vision in two areas. Advanced driving simulation will supply an essential component at least for virtual prototyping of ground vehicles, and the joint ARPA-Army demonstration of the virtual proving ground will benchmark the capability of distributed simulation to support the acquisition process in a quantifiable manner. Most importantly, this work embeds the user/operator in all stages of concept development, test and evaluation as an active participant.

REFERENCES

1. Freeman, J.S., Haug, E.J., Kuhl, J.G., and Tsai, F.F. (1993). "Dynamic Simulation for Vehicle Virtual Prototyping." In Periera and Ambrosio (Ed.), *Computer Aided Design of Rigid and Flexible Mechanical Systems*, Kluwer Academic Publishers, The Netherlands, Series E: Applied Sciences, 268, 533.

2. Hwang, R.S., Bae, D.S., Kuhl, J.G., and Haug, E.J. (1990). "Parallel Processing for Real-Time Dynamic System Simulation," *Journal of Mechanical Design, Transactions of the ASME*, 112(4), 520-528.

3. Bae, D.S., and Haug, E.J. (1987). "A Recursive Formulation for Constrained Mechanical Systems, Part II-Closed Loop," *Mechanics of Structures and Machines*, 15(4).

4. Backker, E., Nyborn, L., and Pacejka, H.B. (1987). "Tire Modeling for Use in Vehicle Dynamics Studies," *SAE Paper No. 870421.*

5. Negrut, D., and Freeman, J.S. (1994). "Dynamic Tire Modeling for Application With Vehicle Simulation Incorporating Terrain," *SAE Paper No. 940223.*

6. Evans, D.F. (1992). "Correlated Database Generation for Driving Simulators," *1992 Image VI Conference Proceedings*, The Image Society, Inc., Tempe, AZ, 353-361.

7. Kuhl, J.G., Papelis, Y.E., and Romano, R.A. (1992). "An Open Software Architecture for Operator-in-the-Loop

Simulator Design and Integration." In Haug (Ed.), *Concurrent Engineering: Tools and Technologies for Mechanical System Design*, Springer-Verlag, Heidelberg, Series F: Computer and System Sciences.

8. *Advanced Distributed Simulation Technology - System Definition Document*, (1992). LORAL ADST Program Office, Orlando, FL.

ADA STRUCTURAL MODELING DESIGN EXPERIENCE FROM AN ENGINEERING MANAGEMENT PERSPECTIVE

T. Michael Moriarty
AAI Corporation
Baltimore, Maryland

ABSTRACT

Ada Structural Modeling (ASM) is a software development concept that emphasizes the architectural aspects of a real-time software system. The concept was developed by the Aeronautical Systems Center, Wright-Patterson Air Force Base, with assistance from the Software Engineering Institute, Carnegie Mellon University. The concept was originally developed for flight trainers, but has recently been used to design the Simulator for Electronic Combat Training (SECT), a high-fidelity, classroom trainer used to train Air Force Electronic Warfare Officers.

As might be expected, the infusion of a new technological concept presented the development team with numerous technical challenges and opportunities. While the specific technical responses to those demands are of interest to the design analyst, the effect of the responses on the program is of interest to the engineering manager. This paper reviews the ASM design experience on SECT, summarizes its effects on the program, and documents lessons learned for using ASM concepts on future programs.

About the Author

Mr. Michael Moriarity is a Principal Development Engineer in the Training and Test Systems business unit at AAI Corporation. He is currently the project engineer on the Simulator for Electronic Combat Training (SECT) program. Mr. Moriarity holds a Bachelor of Science degree in Mathematics from the University of Minnesota and a Master of Engineering Administration from George Washington University. Previous to his current position he was responsible for software development on the A-6E SWIP Weapon System Trainer, EF-111A Operational Flight Trainer, Navy Electronic Warfare Trainer System, and the defensive instructional subsystem on the B-52 Weapons System Trainer.

ADA STRUCTURAL MODELING DESIGN EXPERIENCE FROM AN ENGINEERING MANAGEMENT PERSPECTIVE

T. Michael Moriarity
AAI Corporation
Baltimore, Maryland

BACKGROUND

Structural Modeling

The structural model concept was developed through the collaboration of software engineers from the Air Force Aeronautical System Center (ASC), the Carnegie Mellon University Software Engineering Institute (SEI), and several training system contractors. The effort was started in 1986 and has developed into a working concept that emphasizes structural aspects of a software system. The concept is documented in a number of informal white papers and was the subject of a tutorial at the 1992 I/ITSEC conference. The concept has been used on several large Air Force trainer programs including the B-2 Weapons Systems Trainer, the C-17 Aircrew Training System, and the Special Operations Forces Aircrew Training System. It is currently being used on the Simulator for Electronic Combat Training (SECT) system. This paper describes the SECT project's design experience with structural modeling.

The development of structural models was motivated by the desire to minimize software maintenance costs while addressing users' training needs in an environment of increasingly complex aircraft and training missions and, also, to respond to users' requests for modifications to the training system. A structural model is defined as an object-based application framework, developed at the design level. Object-based means that the design elements within the application framework, the training system software, are derived using the concepts of objected oriented analysis and design. Developed at the design level means that the design element specifications are partitioned to code units and are defined to a level of detail such that implementation, or coding, can begin. Because trainer systems using structural models to date have used the Ada language, the design level has usually taken the form of compilable Ada specifications and Ada based Program Design Language (PDL). Also, because Ada has been used in all of the structural model implementations the concept has often been called Ada Structural Modeling (ASM).

In its simplest form, ASM defines a small set of structural elements that can be used repeatedly to define the structure of the overall system software. The structural elements are defined for two broad areas or layers: an executive layer and an application layer. The executive layer provides the general coordination services required by trainer system software. This includes the sequencing of periodic processing and the dispatching of aperiodic events. Structural elements provided in the executive layer include sequencing routines, calling order lists, and queue handling logic. The application layer contains the trainer software subsystems that perform the unique trainer simulation tasks such as the simulation of flight controls, engines, avionics, etc. All of the application level subsystems are identical to each other in terms of structure and are made up of structural elements such as subsystem controllers, subsystem objects, import areas, and export areas.

Another important facet to ASM is the partitioning strategy used to decompose the training system software into its application level subsystems. ASM uses object oriented design techniques to map application level subsystems and their objects to real-world counterparts. For example, an altimeter application level subsystem with its associated pneumatic, electrical, and indicator objects has a direct correspondence to a real-world altimeter subsystem. It is thought that by closely mapping the software subsystems to their real-world counterparts, changes in the real-world systems will result in changes only in the changed object. Also, proper mapping can reduce the complexity of the simulation problem. In the case of the altimeter system with its objects, the loss of electrical power or the activation pneumatic, electrical, or indicator malfunctions can be easily simulated.

Proponents of ASM believe that the repetitive use of a small number of structures improves integratability and maintainability of a software system. Likewise, a partitioning strategy based on real-world objects improves trainer system software functionality and provides a means of coping with the increased complexity of hardware and software technology. ASM, it is believed, through its reduced set of structures and real-world based partitioning also greatly facilitates the communication of design information throughout the system life cycle.

The SECT Program

SECT is a electronic combat mission simulator used to accomplish primary and mission simulator training in the Air Force Air Education and Training Command's (AETC) Electronic Warfare Officer Training Course. Primary training includes threat recognition, analysis, and exploitation using simulated electronic combat equipment. Mission training includes penetration, standoff jamming/direct support, electronic intelligence collection, and suppression of enemy air defense operations. SECT consists of six student stations in which simulated electronic combat equipment panels are displayed on CRTs. The student interacts with the equipment panels via touch screen controls and is able to receive signals, display them on analysis equipment, and counter them through the use of jamming or expendables. SECT also has a two-position Instructor Console and a Training System Support Center. Figure 1 is a layout of the SECT system. The SECT program is managed by ASC, and the

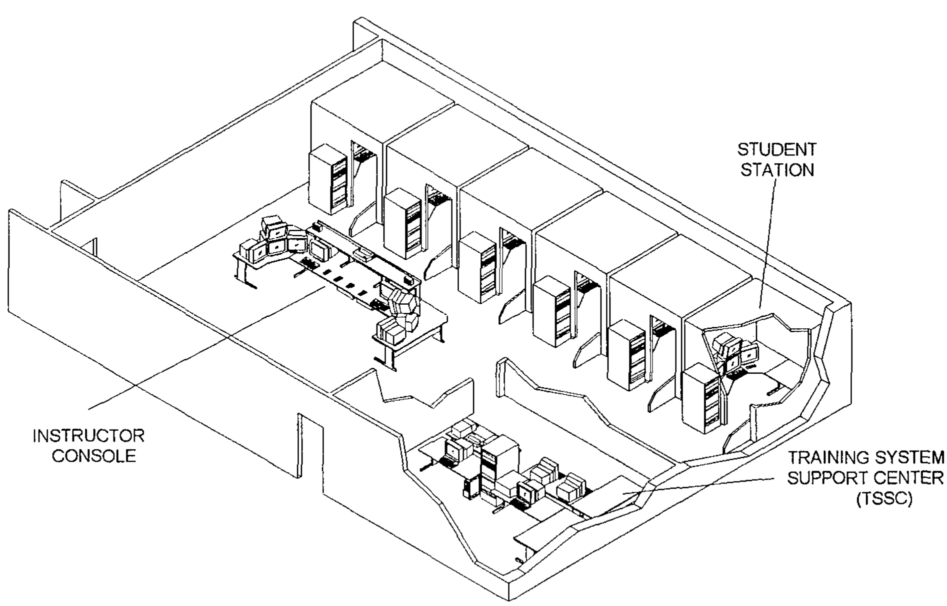


Figure 1. Layout of the SECT Training System

statement of work requires the use of "object oriented methods and Ada Structural Modeling."

Shortly after the award of SECT, an ASM Working Group was established, with representatives from ASC engineering, SEI, and the AAI project team. This working group was the primary means of transferring ASM technology as developed on previous programs to the AAI project team. Sessions early in the SECT program consisted of tutorials presented by ASC and SEI engineers and reviews and discussions of existing white papers. Later sessions consisted of working sessions where specific SECT related design questions were posed, alternatives considered, and design approaches selected. Toward the end of the detailed design phase, as the Critical Design Review (CDR) approached, members of this group participated in subsystem design walk-throughs.

At the time the SECT contract was awarded, ASM techniques had been used on several other large trainer programs. In each of the earlier trainer programs ASM was developed around a flight simulator. SECT represents a departure from these earlier models in that, while SECT includes the concept of a student flight platform or ownship, the student ownship is of secondary importance. The primary entities in SECT are the electronic combat environment and the student's simulated electronic combat equipments. The ownship flight model in SECT is quite simple and basically provides the student with a position and an attitude in the electronic combat gaming area. The aerodynamic handling properties of the simulated ownship are unimportant as long as the ownship's speed, turn rate, and climb/dive rate are similar to the training mission aircraft.

SECT Ada Structural Modeling

The SECT ASM was developed from the earlier flight simulator based ASM efforts. The major elements from the earlier models were retained and relatively minor changes were made for the electronic combat trainer. The concept of the executive layer and application layer is unchanged. The executive layer is made up of an executive and event queue. The executive determines the calling order of the application level subsystems and the rate at which those subsystems are called. The executive, depending on the training system mode (initialize, run, freeze, reset, etc.), calls a predefined list of subsystem controller entries. The event queue provides logic to queue up asynchronous events, or commands, and calls the event processor entry of the appropriate subsystem.

The application layer is made up of a number of subsystems with identical structures. Each subsystem has a subsystem controller which in turn controls the execution of the objects specific to the subsystem. Subsystem control is exercised through two periodic entries (import and update) and/or three aperiodic entries (initialize, reconfigure, and process event). Each of the objects has corresponding entries which are called by its system controller.

Interfaces among subsystems are restricted to two constructs: import/export areas and events. Import/export areas are used largely for the passing of periodically updated data among subsystems. Subsystems that create data place the data in the subsystem's export area. Subsystems that use data have an import element that accesses ("withs" in Ada terminology) the export areas of interest and uses the data needed. Events are used for asynchronous actions and often take the form of a command; are generated by the lesson script, instructor actions, or other subsystems; and generally pass little data and cause an immediate action to occur.

Figure 2 shows a simplified diagram showing the major components of the SECT ASM using a chaff/flare dispenser as a representative equipment model subsystem.

A major extension to the SECT ASM over previously developed models is the use of two executives running at two different update rates. Flight simulators, as compared to electronic combat trainers, are made up of a large number of subsystems that require a moderate iteration rate, say 20 Hz, and each subsystem requires a low to moderate amount of computational processing. The iteration rates for these subsystems, which typically model flight controls, platform motion, avionic responses, etc., need to be sufficiently high to provide realistic responses to student/pilot control inputs. Often, the computational requirements for each subsystem is low to moderate, because the student/pilot can only provide a limited number of inputs to the system in a given iteration. It is noted, of course, that certain flight trainer subsystems such as visual systems or radar landmass simulators do not necessarily follow these generalities.The characteristics of flight simulators generally lead to a single executive structure where many subsystems are processed within the period of one frame and the iteration rate of a subsystem is determined by the number of frames per second in which the subsystem is executed. For example, in a trainer with a basic iteration of 20 Hz all processing, including required spare processing time, has to be accomplished in a 50 msec interval or frame. Subsystems that require a 20 Hz update rate are executed each frame, those that require a 10 Hz update rate are executed every other frame, and those that require a 1 Hz update rate are executed once every 20 frames.

A difficulty with this process is that subsystems that require a 1 Hz update rate but cannot be allocated sufficient time within a given frame need to be time sliced; that is, the task must be subdivided to run over several frames but the subsystem as a whole is executed only once per second. The normal flight simulator often has a large number of subsystems that have to run at the higher iteration rates and time slicing is often minimized. This allows flight simulators to use single rate executives rather efficiently and all processing is performed at the basic iteration rate or some evenly divisible integer divisor of the iteration rate, such as 10, 5, 4, 2 or 1 Hz in the case of a 20 Hz executive.

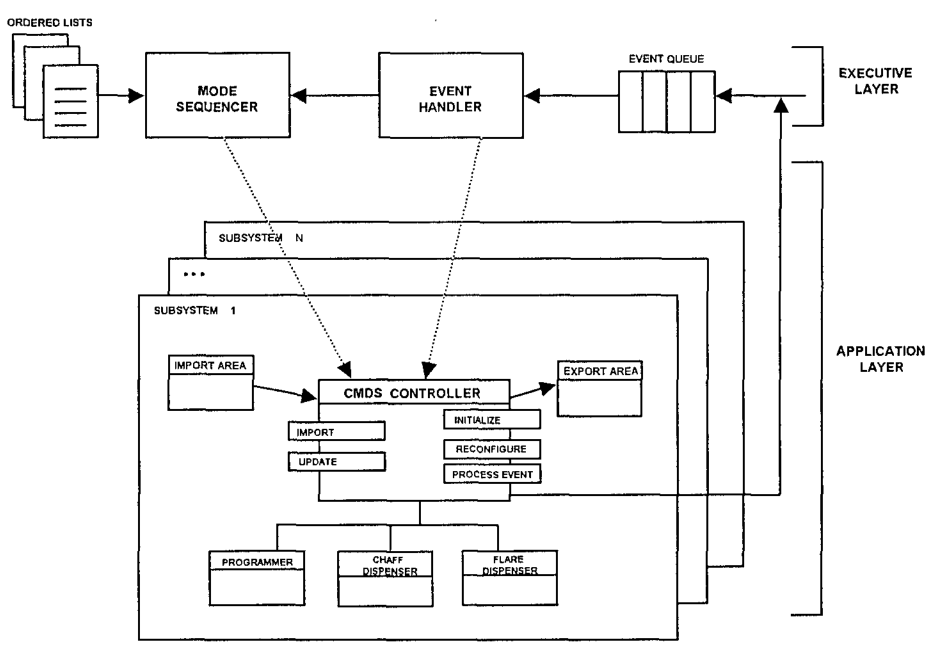


Figure 2 - SECT's ADA Structural Model

The requirements of an electronic combat trainer are somewhat different from those of a flight simulator. The simulation of electronic combat equipment is similar to a flight trainer's simulation of avionic subsystems. The electronic combat equipment subsystems are executed at moderate iteration rates and often have low to moderate computational requirements. These subsystems, like the flight simulator, must respond in a realistic manner to student inputs. The electronic combat environment (ECE), however, is a different matter. Typically, the changes in the ECE occur relatively slowly. New signals coming into view, mode changes on a radar system, and range effects due to changing distance between transmitters and receivers all happen relatively slowly. Updates of 1 Hz or less are usually sufficient for these types of simulations. After a change does occur, however, a great deal of computational processing is required. This type of simulation is best supported by an executive running at a slow update rate such as 1 Hz. At 1 Hz the update rate more closely matches the real-world requirements and the 1000 msec interval allows a greater portion of the processing to be completed so that time slicing does not have to be performed.

The dual iteration requirement resulted in SECT providing a 1 HZ and a 20 Hz executive. This decision added complexity to the executive layer in that additional logic had to be designed to allow for event queues to be processed by both executives. Also, coordination logic had be provided to ensure data integrity in instances that data created by a subsystem called by one executive was valid when used by a subsystem called by the other executive.

SECT'S EXPERIENCE WITH ASM

ASM Learning Curve

The time to learn ASM concepts and use them to develop the SECT's detail designs took longer than estimated. The AAI project team originally based its estimates on a two-step design process: first, acquire a working knowledge of the ASM concepts and techniques, and, second, use the techniques to design the training system software. It was thought that after familiarity with the new techniques was achieved, the remaining design activity was simply to apply the new techniques to a familiar problem. The effort, it was thought, would be similar to replacing a functional design process with an object oriented design process.

The flaw in this thinking was that the ASM design process required at least three steps: first, acquire a working knowledge of the ASM concepts and techniques, second, use the techniques to design the ASM structural elements, and, third, design the trainer software system using only those defined structural elements. The key issue impacting schedule was that two largely sequential design activities were required when applying ASM to a new software domain. The structural elements needed to be designed before the software system could be designed.

Structural Design Validation

One method to validate the structural design is to actually design a portion of the system that encompasses each of the unique aspects of the system. Furthermore, to validate the structural elements, it is important to prototype key portions of the system. This should be done early in the detail design phase so that the structural model can be stabilized before the majority of the detail design is started.

The lesson learned by the SECT project was two-fold. One, get prototyping done early; and two, carefully choose the portion of the system to be implemented for validation purposes. By completing the prototyping early, there will be minimal design effort complete; and, therefore, minimal design changes resulting from structural model changes. Carefully choosing the validation cases can reduce effort and pay high benefits. It is important to pick cases that validate the key system wide control concepts rather than subsystem unique problems.

System Visibility

ASM provided excellent visibility of the trainer system software structure after the structural elements were defined. Because the software system was built up from replicated structural elements, it was easy to get a grasp on the overall system structure. The increased visibiiity was evident when new analysts were assigned to the project or when existing project personnel were assigned new tasks. In the first instance, new programmers, after they understood the nature of the structural elements, developed a good working understanding of how the whole system worked from a subsystem control and interface standpoint. The benefit was even greater when existing project personnel were assigned new systems; they already understood the structure of their new subsystems.

It should be noted, however, that just because an analyst understands the structure of the system, it doesn't necessarily mean they understand the functionality of system. It is one thing to understand how the chaff/flare subsystem is called and how communication is performed, but another thing to realize that the subsystem exists, to know what data it provides or the algorithms it uses.

System Issues Addressed Early

ASM tends to surface high level system control and interfacing problems early. This, of course, is desirable in that it allows correction of these problems early and minimizes the effect on the overall system. The SECT project team uncovered several system level problems early, including a potential data integrity problem that was corrected through a change to the structural model. Had this problem surfaced during integration, it probably would have been difficult to track down and very difficult to correct. A modification to all subsystems would probably have been required. Further, it is possible that, at that late point in a development cycle with its attendant schedule pressures, a compromised correction could have resulted in a less than optimal change that would have impacted system modifiability and maintainability in the future.

The flip side of identifying problems early is that the design phase takes longer and the software staff, to say nothing of their managers, can feel somewhat demoralized in that the design seems never to get done--we keep finding problems.

Software Reuse

Software reuse from previous trainer systems was much less than originally planned. The SECT system was similar to other trainers AAI had built in the past and, while it was realized that the software from these other trainers would have to be rewritten to meet Ada and ASM constraints, it was thought that significant benefits could be garnered from the earlier developed software. The SECT experience has been that while some specific algorithms developed on other trainers were useful as resource data, no software could be reused on SECT from other trainers. The difference between the earlier functionally developed software and ASM requirements was so great that it was determined that the effort to convert old software was greater than the effort to design new software.

In contrast, reusing software developed for one part of SECT on another part of the trainer appears to be easier under ASM than on previous systems. There were several subsystems or parts of subsystems that were developed for use in the student station that could be used in the Instructor Console or in the Trainer System Support Center. Because the calling sequences and the communication methods were tightly controlled under ASM, it was possible to use the same subsystem to perform similar functions in different trainer stations.

Emphasis on Interface Structure, Not Interface Data

The intrinsic emphasis of ASM is to focus on the structure of the software system. A potential negative aspect of this focus is that other important design issues may not get the emphasis they require. One area where this is prone to happen is in the data structures in the import/export areas. ASM concepts put emphasis in identifying the structural elements by which data is transferred and the coordination of those transfers. Likewise object oriented design, to which ASM is related, emphasizes data hiding. The emphasis on structural aspects of data transfer and on subsystem independence through information hiding can result in the lack of proper attention to the definition of the actual data and its structure that is to be passed from subsystem to subsystem. In SECT this happened to some degree in that design walk-throughs paid considerable attention to how data was passed across interfaces and the kind of data that had to be passed, but not to the structure of the data. As a result, additional definition had to be accomplished prior to the start of coding.

Prototyping

The need to define structural elements early and then validate these structures by using them in portions of the system design resulted in prototyping key aspects of the trainer system software early in the design effort. The early prototyping was very beneficial from several aspects. First and most important, prototyping uncovered flaws in the original designs that were able to be corrected early in the development cycle. In many instances the flaw was corrected before it impacted other designs; in some instances designs were already under way and changes had to be made. These flaws were uncovered in the design of structural elements which in turn affected all subsystems in design. In other cases the changes uncovered flaws in the way the structural elements were used in a specific subsystem. In these cases a body of knowledge was built up that other subsystem designers could use in their designs.

Prototyping was also invaluable in that it gave the project team experience in using the software development system early in the program. Development system tools, responses, and resources were exercised and monitored in a realistic manner. Configuration management, archiving, and various housekeeping processes were also verified.

The prototyping effort evolved into a basic test harness for unit testing and later into an integration harness. The basic prototype system consisted of most of the executive layer and several of the simpler subsystems in the application layer. The prototype system supported two different rate executives, interprocessor memory management, event queue logic, and all trainer modes. The application layer had all subsystems stubbed out and several subsystems completed. The prototype system was demonstrated to the customer at various times throughout the development cycle. The most extensive demonstration was at CDR, which strengthened the emphasis on the user aspects of the system

Preliminary Design Review Effects

The emphasis during Preliminary Design Review (PDR) and Critical Design Review (CDR) changed from traditional non-ASM projects. During the SECT PDR the primary focus was the design of the structural model and its structural elements. An important, secondary focus was the partitioning of the trainer system software into its subsystems. A third focus was the identification and development of modeling algorithms to be used to accomplish the simulation. The priority of these three tasks was dictated by the needs of the ASM design process itself--the structure needs to be defined before the subsystems can be defined, and the subsystems need to be defined before the algorithms can be identified.

This had two effects at PDR. The first is that the relative emphasis of the design disclosure met the expectations and needs of the ASC software engineering staff, but did not provide the user with the design information they were looking for. The user's interest was almost the opposite of the engineering staff's. The user would have liked to reverse the priority order--algorithms, subsystem partitioning, and structure.

The second effect was that flaws in the application of structural model concepts to the trainer system software design affected the partitioning of the trainer system, which in turn affected the selection of subsystem algorithms. These issues were worked during the detailed design phase.

In retrospect, a two-step preliminary design review would have worked better. The first review would have reviewed the structural model and its elements. This review would have been of primary interest to the ASC software engineers. A second review would have covered the partitioning and algorithmic aspects of the trainer system software. This review would have been of equal interest to the engineering and user communities.

Critical Design Review Effects

The emphasis during CDR was on trainer system design aspects from the user's point of view. Two circumstances encouraged this approach. The first was simply to provide the user with design data that was needed to ascertain that the system met all training needs. The user emphasis at CDR helped compensate for the information not attained at PDR. The second, and more important aspect, was that the ASC engineering staff already had a good understanding of the trainer system software design before CDR. Between PDR and CDR the SECT system structural model was documented in an "Ada Structural Model Report", all subsystems were designed in strict adherence to the documented model, and the Air Force software engineers were invited to sit in on the internal design walk-throughs. This active participation by Air Force engineers enabled Air Force concerns to be identified early and addressed by the SECT design team in a timely manner. This process also gave the Air Force engineering staff a very high degree of visibility into the SECT design as it was unfolding. By the time the formal CDR was held, the detailed aspects of the software design had already been disclosed to those who had the greatest interest in that design. The resulting CDR then had a greater appeal to a wider range of participants.

SUMMARY

SECT's experience with ASM during the project's design phase raises some concerns but confirms that the goals of ASM are achievable. The concerns raised are the increased time to accomplish system design and the potential difficulty in reusing previously developed non-ASM software. The increase in the design schedule experienced by SECT was related to the need to learn a new design process, to extend the structural model concept into a new class of trainers, and to validate the extended structures. It is anticipated that any design team unfamiliar with ASM concepts and applying the concepts to a new class of trainers would experience similar results. A team, having once gone through the process, and using a previously developed structural model, should experience design times equal to, or better than, those achieved using more traditional design techniques.

The inability to capitalize on previous software designs was disappointing, but in view of the current state of software reusability, perhaps not unexpected. The formalized structure imposed by ASM does reduce the probability that previous non-ASM designs will be usable in ASM projects. The good news is that based on reusability within SECT, there is greater hope for reusability among ASM projects that use common structures.

SECT's experience shows that ASM does improve system visibility and that the increased visibility enables system level design flaws to be identified early in the design cycle. The increased visibility improves communications among designers and between designers and reviewers. The fact that ASM strongly encourages prototyping increases a deeper understanding of the evolving system. It is anticipated that the higher level of visibility will make future maintenance and modification activities more efficient in the ASM based system.

The increased design time and increased visibility into the system software structure does suggest that the content and timing of formal reviews such as PDR and CDR should be revisited. For projects that are developing new ASM structures or are applying ASM to a new class of trainers, an early, formal review of the structural model is recommended. This review should be followed by a more traditional PDR that looks at partitioning and algorithmic designs. Projects that have an active, ongoing ASM working group throughout the design phase should consider shifting the emphasis from detailed software designs at formal CDR to broader more functional design disclosure.

PERFORMANCE LIMITATIONS OF THE DIS INTERFACE

Rodney A. Long, Eric E. Anschuetz, & Lawrence R. Smith
Naval Air Warfare Center, Training Systems Division
Orlando, Florida

ABSTRACT

Distributed Interactive Simulation (DIS) Standards are being established to allow for connectivity and interoperability of dispersed simulations through the standardization of application layer protocols. However, the underlying datagram design is governed by the network bandwidth thus limiting what information can be shared between simulations. The finite bandwidth of serial networks limits how much information can be transferred from one point to another within a specified period. In addition, interfacing to a DIS environment requires a computational element capable of filtering information needed by the individual simulator and performing common functions necessary to interact in this distributed environment. Filtering of simulation data is required since most PDUs are transmitted using broadcast addressing. Dead reckoning provides an engineering tradeoff which reduces network bandwidth, but increases the computation necessary at the simulation interface.

Functions like filtering, dead reckoning, simulation management, collision detection, and time stamping are performed at the DIS interface. The time required to accomplish these functions as well as reliable Ethernet and FDDI communication for DIS is deterministic. The purpose of this paper is to identify the performance limitations of accomplishing the DIS interface as well as to identify the time required to perform the basic functions that make up the DIS interface.

ABOUT THE AUTHORS

Rodney A. Long is a computer engineer at Naval Air Warfare Center Training Systems Division (NAWCTSD) in Orlando, Florida. He graduated from the University of South Carolina with a B.S. in Computer Engineering. He currently works in the NAWCTSD DIS Technologies Group, focusing on the DIS interface area. He can be reached at NAWCTSD, 12350 Reasearch Parkway, Orlando, FL, 32826 or (407) 380-4110.

Eric E. Anschuetz is a computer engineer at NAWCTSD. He graduated from Eastern Michigan University with B.S. degrees in Computer Science and Mathematics. His current duties include software development for the DIS Network Interface Unit (NIU). He was also a participant in the 1993 I/ITSEC DIS demonstration.

Lawrence R. Smith is an electronics engineer in the NAWCTSD DIS Technologies Group. He received his B.S. in aerospace engineering at the U.S. Naval Academy in 1980 and his M.S. in aeronautical engineering at the Naval Postgraduate School in 1988.

PERFORMANCE LIMITATIONS OF THE DIS INTERFACE

Rodney A. Long, Eric E. Anschuetz, & Lawrence R. Smith
Naval Air Warfare Center, Training Systems Division
Orlando, Florida

INTRODUCTION

Distributed Interactive Simulation (DIS) Standards are being established to allow for the interoperability of disparate simulation systems through the standardization of application layer protocols. Through the use of the DIS communications architecture, individual systems can be interconnected to allow real-time, mutual interaction in a common synthetic environment. DIS protocols are used to transmit the minimum amount of information necessary to represent "ground truth." Using this data, each individual system is responsible for determining its own perception of the environment.

Unfortunately, today's technology contains certain bottlenecks that limit the amount of data that can be exchanged in real-time. These constraints include network capacity between systems, as well as the computational load on the individual systems. As a result, engineering tradeoffs are being made which reduce network bandwidth requirements, but increase the computational load at the simulation's network interface.

THE DIS NETWORK

The goal of DIS is to enable *distributed* simulations to interact in a common environment. Therefore, connectivity is a major consideration. Local Area Networks (LANs) can offer data rates from Ethernet's 10 Megabits per second (Mbps) to FDDI's 100 Mbps. However, in moving to a Wide Area Network (WAN), the data rates drop off dramatically due to delays in routers, switches, and the physical transmission medium. T1 networks are the most widely-installed WANs and offer a maximum data rate of only 1.54 Mbps.

Network Limits

Many factors influence DIS bandwidth during a networked exercise. These include: total number of entities, mixture of entity types, type of exercise or scenario, choice of dead reckoning algorithm (including positional and angular thresholds), and security requirements. Presently, the majority of network traffic involves Entity State Protocol Data Units (PDUs). [1]

An Entity State PDU provides specific information such as entity type, location, velocity, orientation and dead reckoning parameters. These PDUs are required to be sent at some minimum rate (e.g., every 5 seconds) by every entity and may also be sent much more frequently depending on entity dynamics.

A BBN review of DIS network loading from I/ITSEC-93 showed peak load periods which contained roughly 230 packets per second for 210 entities (generated by 49 applications). Most packets were about 186-200 bytes, including the overhead of UDP, IP and 802.3 header information. At the high end, this equates to approximately 0.37 Mbps. [2]

Filtering was performed by BBN to isolate various hosts. One host sent traffic from 8 live F-15s, accounting for about 20% of the network traffic; 96% Entity State PDUs, 2% Emission, 1% Transmitter, 1% other. As was expected, the fast movers (aircraft) produced the majority of network traffic, consisting mostly of Entity State PDUs.

Using an F-14 simulator from the NASNET program, we performed a variety of maneuvers to determine the worst case PDU production rate. With an update rate of 30 Hz, the system performed dead reckoning using DR algorithm #4 (second order position, first order orientation) with a tolerance of 1 meter for position and 3 degrees for orientation. Performing the "corkscrew," an ascending spiral, the system peaked at 15 Entity State PDUs/s. This is a typical flight pattern used after a bombing run to evade ground fire.

Network bandwidth saturation is a known problem which will affect very large exercises. Ethernet LANs have been observed to congest significantly at around 60% capacity; this is less than 20 times I/ITSEC-93 peak traffic. In comparison to major exercises being planned, I/ITSEC-93 was a rather small DIS exercise involving only a limited subset of PDU traffic. Other PDUs to be used in future exercise scenarios, such as the Emission or Signal PDUs, may have a more substantial impact on bandwidth than the Entity State PDU.

DIS NETWORK INTERFACE UNIT

Interfacing to a DIS environment requires a computational element capable of filtering information needed by the individual simulator and performing common functions necessary to interact in the simulated environment.
These functions are currently being defined by the DIS Interface Subgroup in the DIS Interface Functional Requirements Document (FRD).

The Naval Air Warfare Center Training Systems Division (NAWCTSD) has developed a DIS Network Interface Unit (NIU) as part of a Cooperative Research & Development Agreement with Motorola. The NIU performs several of the functions identified in the FRD including filtering of DIS Protocol Data Units (PDUs), dead reckoning, coordinate conversion, time stamp generation, and entity collision detection. These functions are described further in the following paragraphs.

DIS PDU Filtering

Filtering of simulation data is required since most PDUs are transmitted using broadcast addressing. The NIU will filter incoming PDUs to decrease the amount of data being processed. Filtering may be performed by exercise number, PDU type, entity kind, entity domain, entity category and distance from ownship.

Dead Reckoning

Dead reckoning is a method of position/orientation estimation used to reduce transmission of Entity State PDUs. By estimating the position and orientation of other systems' entities, it is not required for the application to receive a report about every change in position/orientation that occurs in the remote entities it is dead reckoning. An update is only required when a change in position/orientation differs by a certain amount from the dead reckoned position/orientation. Dead reckoning provides an engineering tradeoff which reduces network bandwidth, but increases the computation necessary at the simulation interface.

Coordinate Conversion

When existing simulation systems are upgraded for DIS, coordinate conversion is often necessary. The NIU will convert between the DIS standard World Coordinate System (Geocentric) and Topocentric, Universal Transverse Mercator (UTM), or Geodetic. Every incoming and outgoing PDU could undergo a coordinate transformation. The conversion process becomes a tradeoff between precision and computational load; precision is directly related to the amount of computations performed.

Time Stamp Generation

Time stamping is used to indicate the time at which the data in the PDU is valid. The time stamp represents units of time passed since the beginning of the current hour. The NIU uses relative time stamping.

Collision Detection

The NIU compares every local entity position with every remote entity position. When the NIU determines that the position of an outside entity is within a specified distance of any of the local targets, it will issue a Collision PDU.

CPU LOADING

The NIU was implemented as part of the NASNET F-14 simulator. The NIU runs on a Motorola 187 board using a Motorola 88100 RISC CPU (see Table 1 for benchmarks). For synchronization purposes, the NIU was run at 30 Hz, which is a common iteration rate for aircraft simulators.

Our first measurements determined that the major CPU intensive functions were dead reckoning and coordinate conversion. Loading effects due to time stamping, PDU filtering and collision detection were minimal in comparison. Upon further analysis, we realized that the Motorola 187 board did not perform transcendental functions such as sine and cosine in hardware. Since dead reckoning and coordinate conversion are math intensive, a hardware implementation of these functions would significantly improve performance.

SYSTEM	MIPS	SPEC INT 92	SPEC FP 92
88100 (33 Mhz)	50	27.7	18.8
* HP 735	76	52.4	149.8
* HP 750		48.1	75.0

TABLE 1: CPU PERFORMANCE BENCHMARKS

* Other computer systems shown for comparison.

The NIU performs dead reckoning using geocentric coordinates and then converts them to the coordinate system used by the simulation application. As this is a function of the number of entities in the Entity Table, we decided to test for the maximum number of entities that could be dead reckoned and had to undergo coordinateconversion. A Semi-Automated Force (SAF) program generated a variety of entities at a constant velocity, with DR algorithm #2 (first order position, no orientation). Using the UTM coordinate system, the NIU began missing frames at approximately 65 entities.

NIU MEMORY REQUIREMENTS

The NIU stores Entity State information in an Entity Table, which includes data from the Entity State PDU as well as other data such as the current dead reckoned position. The current implementation of the NIU can store approximately 1000 entities using 2 MB of memory. Since the CPU cannot process this amount of entities in real-time, memory does not become a factor.

RECOMMENDATIONS

As can be seen, the network interface can easily become overloaded. There are different approaches to solving the problem, none which are completely satisfactory.

The easiest approach is to wait for technology to build faster and cheaper computers. Given recent advancements in technology, this could be in the near future. However, it does not solve today's problems.

Using multicast addressing, the amount of traffic arriving at the network interface will be greatly reduced. Filtering will be done at the hardware level vice software. However, large exercises can be envisioned where a simulator's "area of concern" would include more entities than it could process.

A final alternative, is to filter the information that is processed for the simulation application. While a high level filter at the network is useful for filtering out network traffic from other exercises, a "world view" filter enables only data which is critical to the simulation application to be processed. Using this method, dead reckoning and coordinate conversion would be performed only on the most critical entities as defined by the application.

CONCLUSION

The Interface Subgroup is investigating the development of test tools and procedures to validate basic interface functions. Using functional definitions, the SubGroup is defining a process to: characterize basic parameters, evaluate interface functions, develop test requirements, validate interface performance, and document DIS interface products' capabilities and limitations.

REFERENCES

[1] Guidance Document (DRAFT), Communication Architecture for Distributed Simulation (CADIS), IST-CR-92-21, November 1992.

[2] Seeger, Joshua Dr. , "Network Oriented Scalability", DIS Workshop, March 1994.

USING BENCHMARKS AND SIMULATOR LOADS FOR MULTI-PROCESSOR COMPUTER SYSTEM EVALUATION

Carl Mickelson, Scott Hill, Steve Scibetta
Loral Defense Systems - Akron
Akron, Ohio, USA 44315-0001

ABSTRACT

Traditionally, the selection of computer systems for simulation has been made on the basis of synthetic benchmarks. Advances in computer technology have caused this traditional method to poorly predict the requirements of full training system loads. Modern, commercial off-the-shelf (COTS) simulation computer systems often include multiple processors, shared memory, time-shared system busses, and coherent multi-level cache memories. These systems are notoriously hard to benchmark since traditional benchmarks fail to accurately model a multi-processor simulation load with respect to cache memory and shared resource utilization.

The technique presented uses computer system theory and round-robin resource contention to consume a known portion of the processing capacity of the system being evaluated. Standard benchmark and partial simulator load programs are then run on both the loaded and the unloaded system to determine the effect of the programs upon the performance of the resource contention generator, and also the effect of the resource contention program upon the performance of the programs.

A software metric was created and validated which consumed a known portion of the multi-processor shared bus system being evaluated, independent of both the instruction and data caches in the system architecture. Use of the metric, together with traditional benchmarks and simulation load programs provided insight into the scalability and ultimate performance of the selected simulation computer system. Use of this technique confirmed that traditional methods are indeed misleading when applied to modern simulation computing systems.

BIOGRAPHIES

Mr. Mickelson is an Engineering Specialist Senior with Loral Defense Systems - Akron, 1210 Massillon Road, Akron, Ohio 44315-0001, (216) 796-8029. He has 18 years experience with Goodyear Aerospace and Loral carrying out a wide variety of job assignments in Computing Systems Architecture Analysis and Digital Systems Design Engineering. Mr. Mickelson holds a Bachelor of Science in Engineering (1967) from Case Institute of Technology and a Master of Science in Computing and Information Sciences (1969) from Case Western Reserve University.

Mr. Hill is a Scientific Analyst with Loral Defense Systems. He has 5 years experience in Loral's Flight Simulator department with an emphasis on simulator executive development. Mr. Hill holds a Bachelor and Master of Science in Electrical Engineering (1988) from Case Western Reserve University.

Mr. Scibetta is a Scientific Analyst with Loral Defense Systems. He has 9 years experience with Goodyear Aerospace and Loral with a range of experience including Training Systems, Radar Simulation, Flight Simulation, and Computing Systems Architecture Implementation. Mr. Scibetta holds a Bachelor of Science in Computer Science (1985) from The University of Akron and a Master of Business Administration in Information Systems (1991) from Case Western Reserve University.

USING BENCHMARKS AND SIMULATOR LOADS FOR MULTI-PROCESSOR COMPUTER SYSTEM EVALUATION

Carl Mickelson, Scott Hill, Steve Scibetta

BACKGROUND

Real-time flight simulators have always stressed the processing capacity of the computing systems used in their implementation. One of the first problems that the simulator engineer must attempt to solve is to estimate the performance of the computing systems that are candidates for use in the simulator to be built. In order to address this problem, various benchmark programs are typically developed to measure the processing capacity of each of these systems. Prior art simulation systems, based upon single processor or master-slave processor computers, have been built successfully using such performance assessment tools. Once developed, these assessment tools are typically re-used on new programs to help determine the capacity of the processing systems needed for the new project.

As new microprocessors have been developed, many different computing system vendors have adopted the same microprocessor as the base technology for their individual value added products. Further, many of these new systems include the hardware and software necessary to support symmetric multi-processing for real-time simulation systems. The problems now confronting simulation systems design engineers include how to intelligently differentiate among the variety of similar computing systems available, and how to best measure the performance of candidate systems using the performance assessment tools with which he has had experience.

The tendency exists to try to scale the use of the familiar performance assessment tools to the new candidate platforms, and extrapolate the results of their use for the number of processors in the candidate system. Because of the way individual benchmarks were developed for single processor environments, it is often not possible to properly estimate the effects of such multi-processor system architectural features such as shared bus access to memory, and cache characteristics such as size and organization. The purpose of this paper is to present a technique that has been developed to characterize the expected performance of the Concurrent Computer Corporation C7550 68040-based multi-processor computing system that has been selected for use in the Special Operations Forces Aircrew Training System (SOF-ATS) flight simulator development effort. The method can be easily applied to other system architectures.

This paper documents the second phase of a two phase effort to characterize the performance of the subject computers identified above. The first phase effort was fully documented in a paper presented at the 1994 ITEC Conference at The Hague. The pertinent conclusions from that effort are presented here, followed by a discussion of how the use of partial simulator program loads can be used to examine system performance.

CACHE BUSTER / BENCHMARK CONCLUSIONS

These conclusions were drawn from the first phase research effort:

- The 7550 round robin bus arbitrator worked properly. From the Cache Buster characterization data it was clear that the bus was correctly sharing resources among client applications and that there were no stray, unaccounted for bus cycles.

- The 68040 caches, although small, were beneficial in all cases. The effect of turning off all caching damaged performance up to an order of magnitude, even for the most cache busting applications.

- In order to linearly scale performance as the number of CPU's in a computing system is increased, it is essential that the processors, their caches and/or their memory systems be locally interconnected. Two of the benchmarks, ELECT5 and FCMOM, demonstrated the significance of such a local interconnection.

- Two of the benchmarks, ETMM5 and FWICG, significantly broke the cache of the 68040. ETMM5 broke the cache more severely than FWICG. Since ETMM5 was the benchmark that was designed most closely as a flight simulator, the performance achieved when using this benchmark probably represented the upper bound on the system performance that can be achieved for the selected architecture.

APPROACH TO PERFORMANCE ASSESSMENT USING PARTIAL SIMULATOR LOADS

After using the Cache Buster metric summarized above to develop an understanding about the benchmark performance of the Concurrent 7550 being used on the SOF-ATS program, an effort was undertaken to validate these insights by testing how this computer system would perform using actual simulation application software. The results of these tests are discussed here, following a description of the different test conditions under which the software was prepared and run, and a discussion of how to compare and interpret the numerical results in the charts in this paper.

Each of the tests was run simulating the same aircraft configuration and flight profile, taxiing down a straight runway at a constant speed of 250 knots, with a straight and level attitude. The purpose of running the tests under these conditions was to involve all the significant software subsystems in the simulation, including aero modeling, ground effects calculations, and landing gear and strut loading computations. It was felt that this would represent a reasonably complex, though somewhat artificial, set of long term flight conditions that could be used to load the simulation software. Such conditions represent a near worst-case scenario for the number of subsystems that will be active during the frame based calculations that take place during simulation.

Different software system configurations were used to gather statistics for the analysis of the Flight Station system performance. While each of these tests applied a different set of conditions to the software being executed, the aircraft and flight conditions being simulated were held constant as explained above. The paragraphs that follow define the conditions under which each of the five 30 Hz test runs and the three 60 Hz test runs were performed. The figures included at the end of the paper present the test results discussed below and summarized in Table 1.

Flight Station (FS), 30 Hz, Baseline Test

This test was run using the software compiled with Ada run-time checks enabled, and function call inlining disabled. This is the same software compilation configuration that has been used during the initial development and integration activity. This baseline test used the same subsystem activation rate table that has been used during development, but excluded all High Speed Device (HSD) sensor and actuator I/O activity to the CPIO intelligent IO hardware system and control loader, and did not schedule any "time burner" subsystems. See Figures 1a through 1d.

FS, 30 Hz, Checks Suppressed (CS)

This test used the same source software as the baseline above, but recompiled with Ada run-time checks suppressed, and function call inlining enabled. The same baseline subsystem activation rate table was also used here to dispatch the individual software subsystems. The rate table is an input file to the simulation executive that defines the specific software subsystems that are to execute in each frame of each one second simulation time interval. By changing the contents of the rate table, the different software subsystems in the simulation can be activated on different frames. This technique can be used to help balance the level of computing activity in each frame as an aid to balancing the computational load across all the frames of the one second scheduling interval. See Figures 2a through 2d.

FS, 30 Hz, CS, Frame Load Balancing (LB)

This third test run made use of the same software version that was used in the second run above, but used a modified rate table that was tailored to try to evenly spread the computational load of the simulation across all frames. It must be noted here that the load balancing performed for this test run, and all other test runs where this load balancing principle was applied, was limited to changing the frames within each scheduling interval where the different subsystems running on that CPU are active. The goal here was to evenly distribute across all frames within a scheduling interval the amount of work each CPU was required to perform. The specific subsystems scheduled to run on a particular CPU were not modified for these test runs. See Figures 3a through 3d.

The ability to achieve further load balancing by switching certain subsystems to the alternate CPU was not exploited here because the computational interdependencies of the different subsystems was not yet fully defined. Once such a definition is completed, it should be possible to switch some of the subsystems to run on the alternate CPU to achieve not only a constant compute time per frame per CPU, but also the same constant time per frame on each CPU. Such a condition would represent a perfectly balanced use of the available computing resources for the duration of the 30 or 60 frames in each scheduling interval.

FS, 30 Hz, CS, LB, 15 millisec Time Burner (15 TB)

This test added a 15 millisecond "time burner" to the subsystem rate table for each of the two 68040 CPU's used by the simulation software. In one CPU of the pair, the time burner task was scheduled to run first, before any of that CPU's software subsystems were scheduled. On the second CPU of the pair, the time burner was scheduled to run after all other scheduled subsystems were run. It was reasoned that, by using the time burners to stagger the timing of the simulation computations in each CPU, this test would minimize the amount of central memory bus contention that would be experienced during the simulation run. The results of running the benchmark tests already discussed had illustrated the performance impact of central bus access contention. This test run was designed to determine if simulator performance could be enhanced by properly scheduling the computational activity in each CPU so that such bus contention is avoided. See Figures 4a through 4d.

FS, 30 Hz, CS, LB, 15 TB, HSD IO

The last 30 Hz test was run after enabling the IO software subsystems in the executive rate table. This run was designed to determine the effect of performing the normal HSD IO operations that would be experienced while the simulator is running after being fully integrated with the cockpit, control loader, and motion systems. However, since no real cockpit, control loader or motion systems were actually connected to the test computer system, all of the data read into the simulation by the I/O subsystems was ignored, so that the flight scenario defined above could be maintained. See Figures 5a through 5d.

FS, 60 Hz, CS, Frame Load Balancing (LB)

This first of three 60 Hz tests duplicated the software compilation and executive scheduling conditions described for the second and third 30 Hz test runs, except that the number of frames scheduled in each one second scheduling interval was raised from 30 to 60. See Figures 6a through 6d.

FS, 60 Hz, CS, LB, 8 millisec Time Burner (8 TB)

This 60 Hz test run used two 8 millisecond time burner subsystems, scheduled one in each CPU, to stagger the computational activity in each CPU and minimize bus contention as in the fourth 30 Hz test run. In all other respects, this test run was identical to the 60 Hz test above. See Figures 7a through 7d.

FS, 60 Hz, CS, LB, HSD IO

This 60 Hz test run added the HSD IO subsystems to the software being scheduled in each frame. In all other respects, this test run was identical to the first 60 Hz test run. See Figures 8a through 8d.

TEST DATA COLLECTION

As each test run was made, the data collected included the duration of each frame within each one second scheduling interval. During the run, the longest frame time for each frame was identified, and the average frame length for each frame across the entire test run was computed. These statistics were collected separately for each CPU in use by the software. Additionally, each frame's elapsed computation time on each CPU was categorized into an eight bin histogram. The intention of the histogram is to determine how stable the computation time of each frame was across the duration of the entire simulation run.

The results of each test run produced a frame time histogram for each CPU, and three additional charts that present average and peak frame times for each CPU. The frame time charts graph the average time for each frame in every scheduling interval taken over the entire simulation run. (For those test runs that applied time burners to stagger the calculations in each CPU, the time charts are plotted with the effect of the fixed length time burner removed, exposing the true calculation time for each average frame.) A number of general observations are pertinent when interpreting these graphs.

The width of each histogram is a measure of the spread between the minimum and maximum frame times, while the shape of the histogram is a measure of how the frame times were distributed throughout the simulation run. An ideal histogram for these test runs would have a single bin populated, indicating a near constant compute time per frame, with little spread between the minimum and maximum frame time.

The ideal graphs of the average and maximum frame times per CPU should show two horizontal lines that overlap each other. Such a graph would indicate that the average and maximum frame times were constant and identical. While this objective is the ideal, indicating a well balanced CPU load, none of the test runs produced such ideal results. Some of the graphs produced from the test runs reported here illustrated large variation in the average time per frame. Such variation indicates that certain frames have a large amount of work scheduled, while adjacent frames have only a small amount of work to perform. This results from a computational load that is not evenly balanced across all available frames. By examining the horizontal distance between the peaks and valleys in such a graph, the frequency of the high load frames can be determined, and depending upon how subsystems are scheduled, the high demand subsystems can be identified. If these high demand subsystems are scheduled on different, rather than the same frames, such peak computational demands can be avoided, and the load evenly distributed across all frames.

To evaluate the validity of the tests that were performed for this paper, the results of the 30 Hz test runs were compared both against the 30 Hz baseline run, and against each preceding test for reasonableness. Table 1 presents a brief summary of the results of all the test runs and the relative improvements encountered as the different test runs were made. The footnotes provide some insights regarding the significance and reasons for the reported differences. The charts on the next two pages present average frame times and the average per frame performance improvement due to the optimizations used in test run 4 above. The top line in each chart (more widely variable for CPU 3) presents the average frame time collected from the 30 Hz baseline test run. The center line plots the average frame times for test run 4 (with the time burner time removed). The bottom graph represents the net performance improvement in milliseconds per average frame due to the combined effects of Ada check suppression and function call inlining, load balancing, and run time staggering by the use of properly scheduled time burners. These charts graphically depict the 37% to 39% improvement in performance identified in the chart above for test run 4 when compared to the baseline. It must be remembered that the beneficial effect of the staggered time burners as the processing demand per CPU exceeds 50% of each frame will be reduced because of the necessary and unavoidable bus contention that will occur as each CPU needs to accomplish additional processing in each frame.

PARTIAL SIMULATION LOAD CONCLUSIONS

The data recorded during these tests, and the results presented above are indicative of the types of performance improvements that can be achieved as the different optimization techniques used here are applied to the SOF-

ATS simulation software. As is described in the footnotes to Table 1, each technique provided results that are both explainable and reasonable. The results of these tests are encouraging, net gains in performance of approximately 30% are easily achievable without refining the source code and processing times of under 8 milliseconds, at 60 Hz, are possible for this reasonably robust exercise of the Flight Station software. Additional performance improvement may be possible after completion of software integration when inter-CPU load balancing can be accomplished, and the software can be examined in more detail to determine whether some source code re-design can be accomplished to minimize the amount of computation being performed in each frame.

Test Case / Figure #		Net Avg (a) (millisec)	Chg from Baseline	Change from Prior Test
1. FS 30Hz Baseline	cpu2	13.761		
	cpu3	14.153		
2. FS 30Hz CS	cpu2	12.167	- 11.6%	- 11.6% (b)
	cpu3	12.167	- 14.0%	- 14.0%
3. FS 30Hz CS LB	cpu2	12.932	- 11.5%	+ 6.3% (c)
	cpu3	12.861	- 9.1%	+ 5.7%
4. FS 30Hz CS LB 15 TB	cpu2	8.596	- 37.5%	- 33.5% (d)
	cpu3	8.525	- 39.7%	- 33.7%
5. FS 30Hz CS LB 15 TB IO	cpu2	10.658	- 22.5%	+ 24.0% (e)
	cpu3	8.046	- 43.1%	- 5.6%
6. FS 60Hz CS LB (60 Hz Baseline)	cpu2	10.277		
	cpu3	6.788		
7. FS 60Hz CS LB 8 TB	cpu2	7.843	- 23.7% (d)	
	cpu3	4.392	- 35.3%	
8. FS 60Hz CS LB IO	cpu2	11.663	+ 13.5% (e)	
	cpu3	6.497	- 4.2%	

(a) Net Average implies subtraction of time burner from captured averages.
(b) Denotes gain from suppressed checks and function inlining.
(c) Due to load balancing to achieve more consistent times from frame to frame, the processes now contend for the bus more than they did when not balanced.
(d) Implementation of time burners helps to eliminate bus contention by allowing each process dedicated access to the bus.
(e) Inclusion of HSD I/O for Control Loading and CPIO, affects cpu2 since these subsystems are run on cpu2. cpu3 gains by having greater access to the bus while cpu2 waits for I/O.

Table 1 - Partial Simulation Load Test Run Comparisons

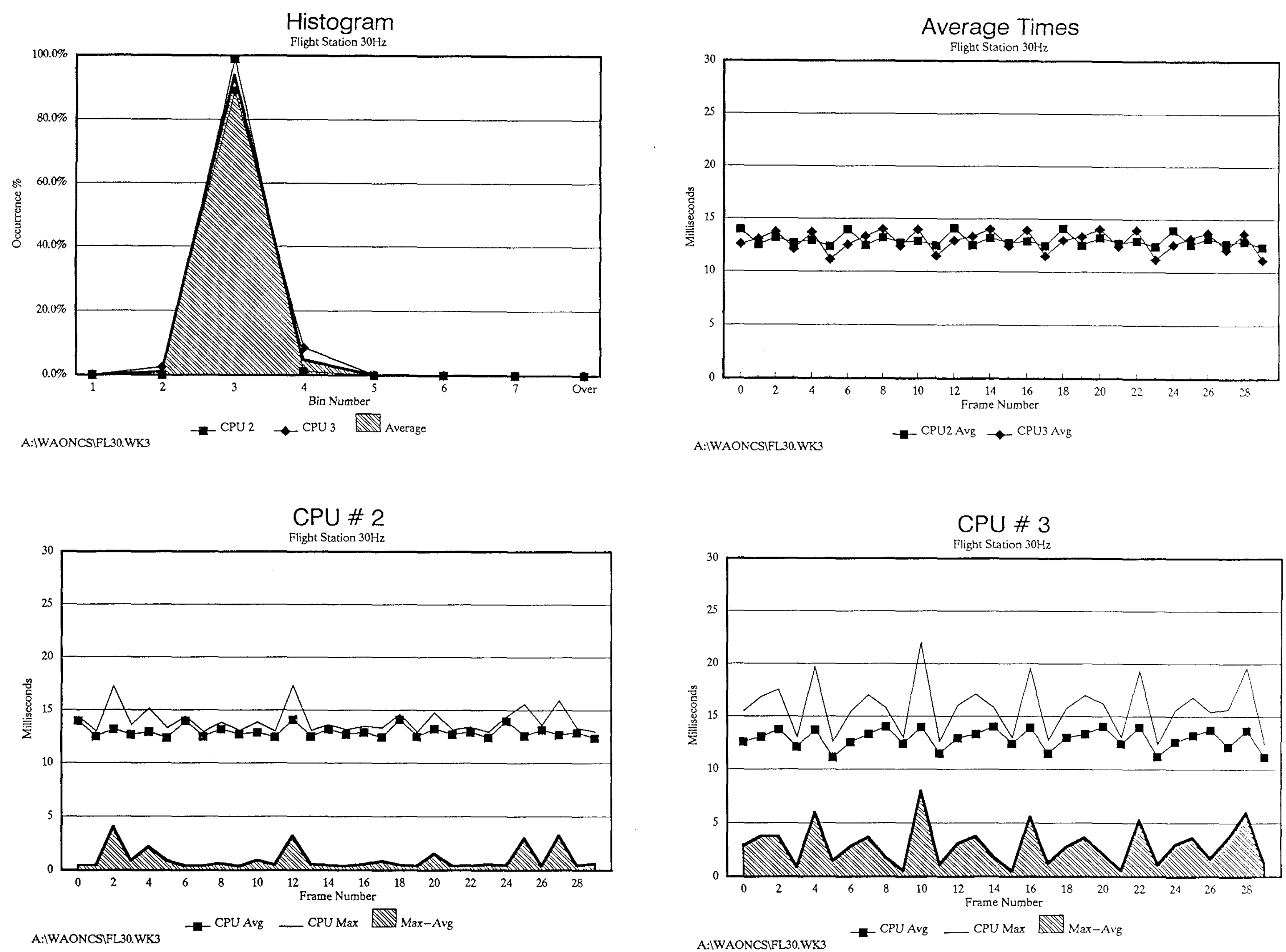


Figure 1 - Flight Station, 30 Hz, Baseline

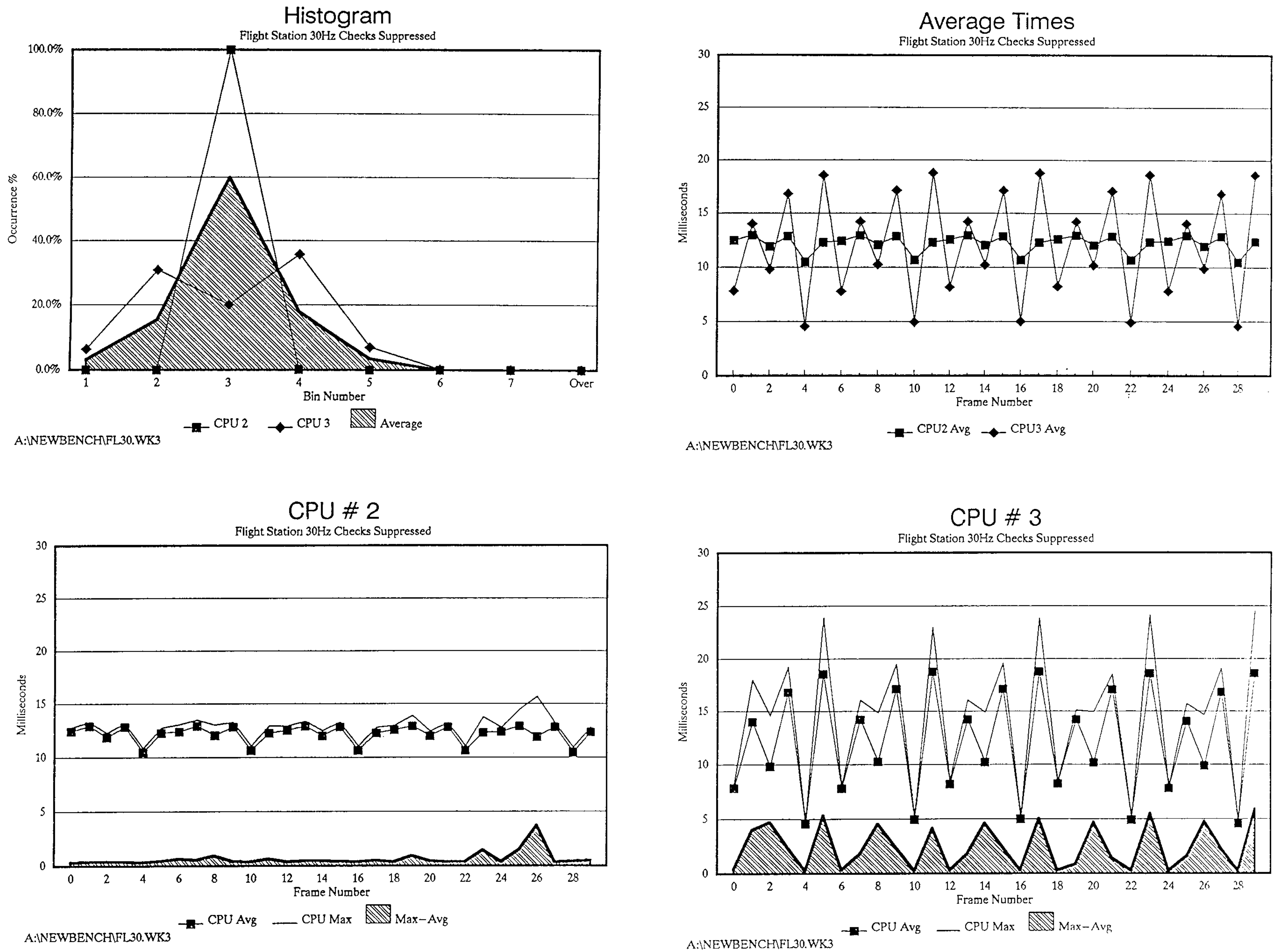


Figure 2 - Flight Station, 30 Hz, Checks Supressed

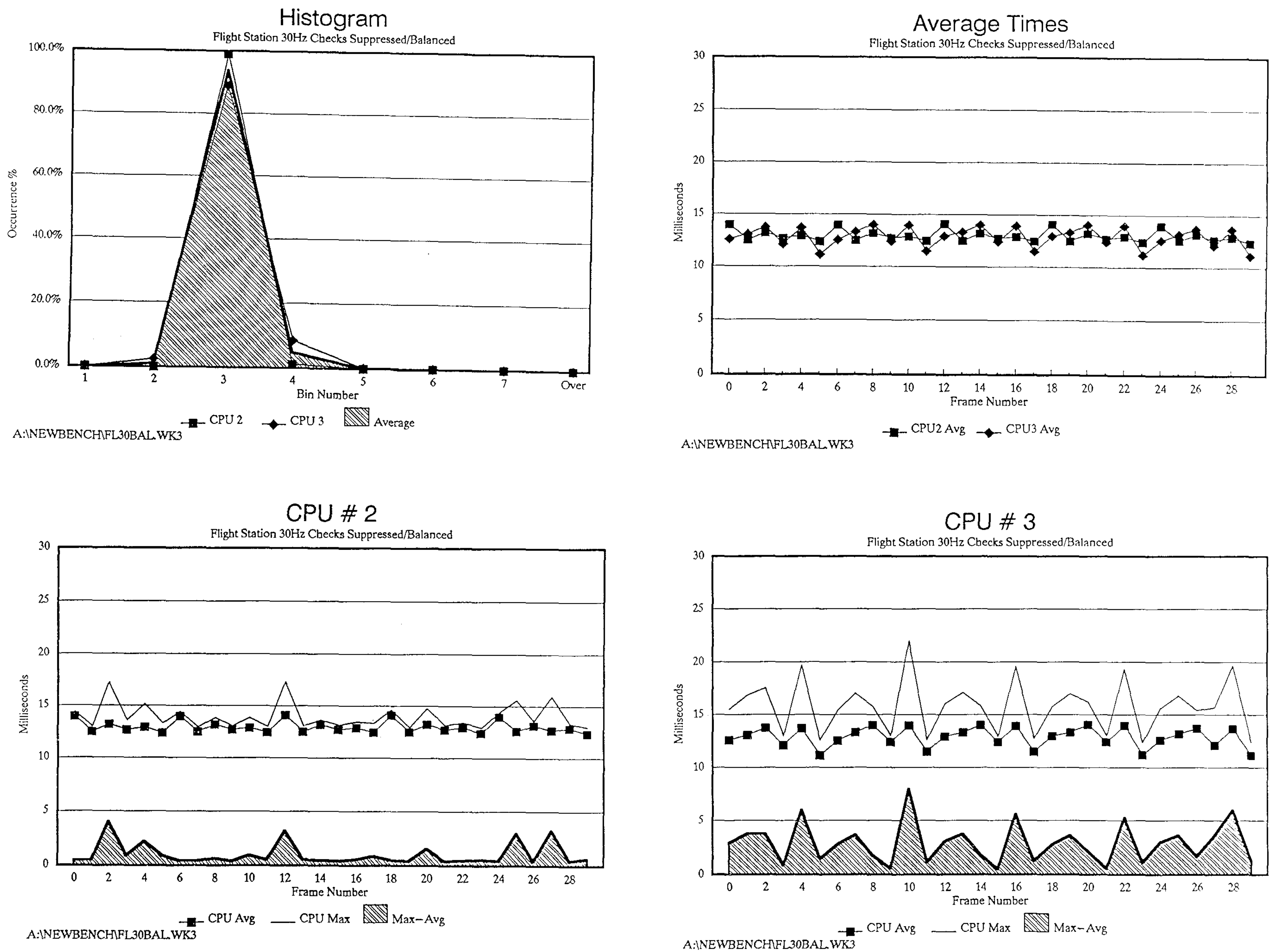


Figure 3 - Flight Station, 30 Hz Checks Supressed, Load Balanced

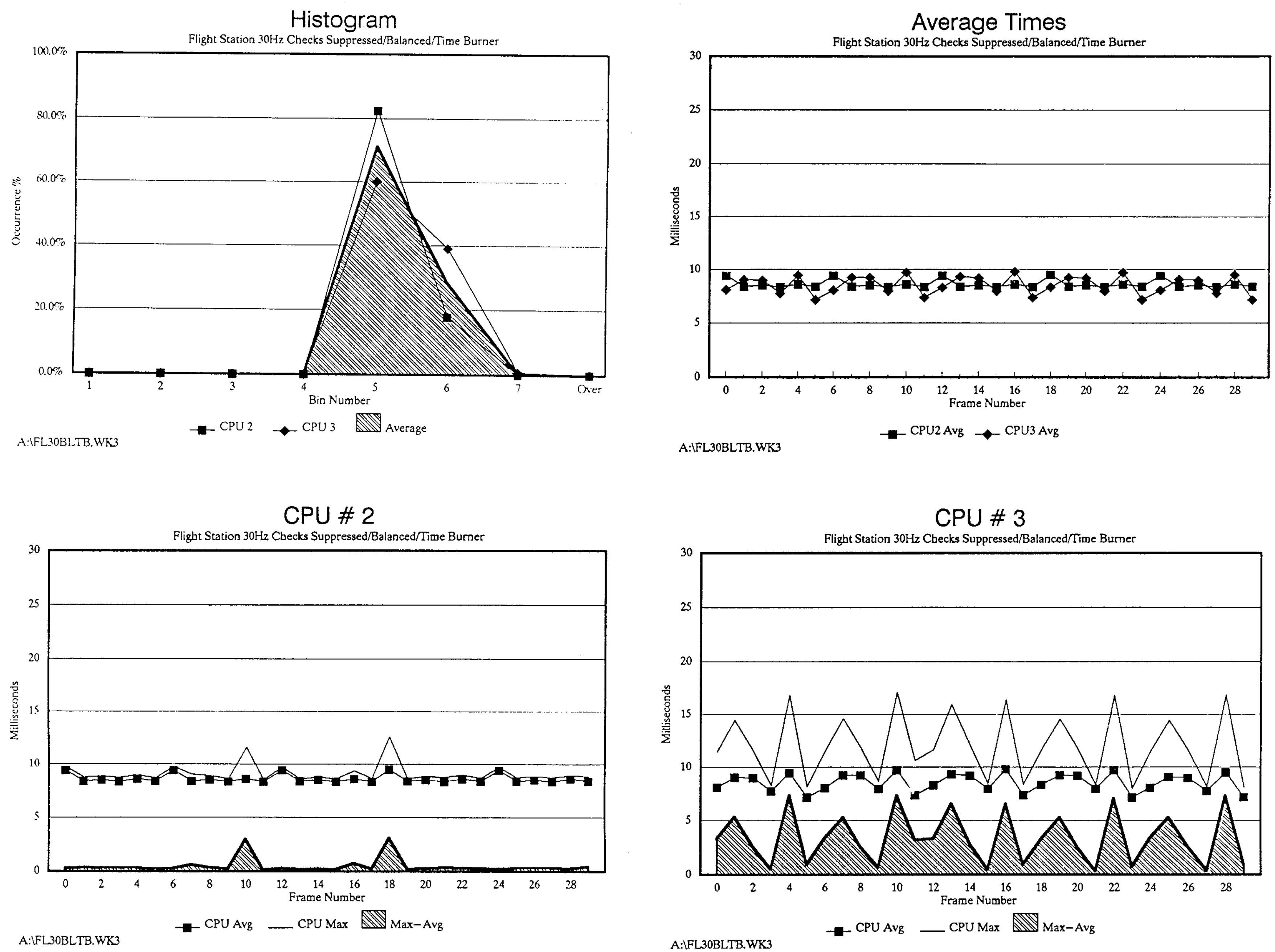


Figure 4 - Flight Station, 30 Hz, Checks Supressed, Load Balanced, 15 millisec Time Burner

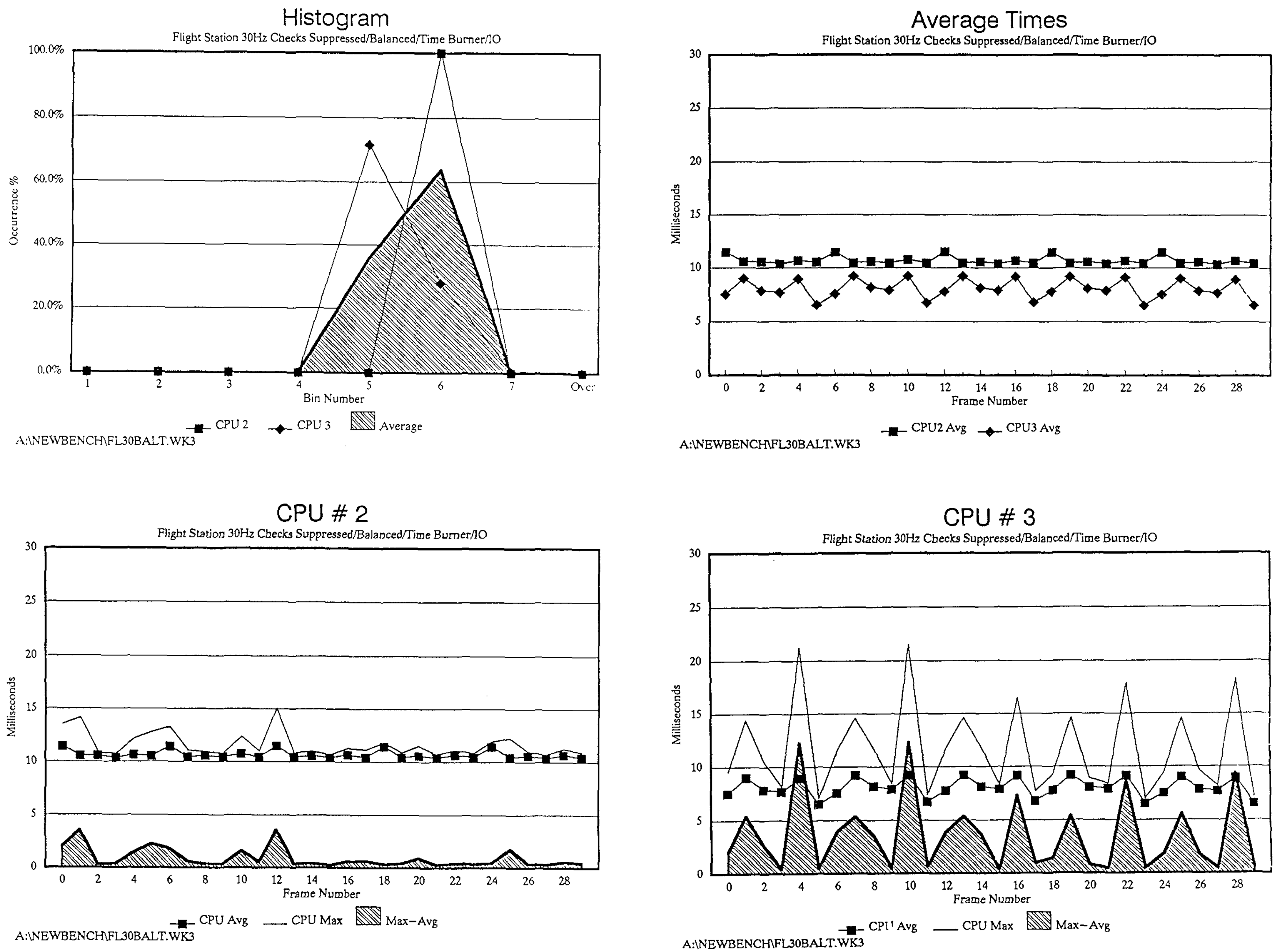


Figure 5 - Flight Station, 30 Hz, Checks Supressed, Load Balanced, 15 millisec Time Burner, HSD IO

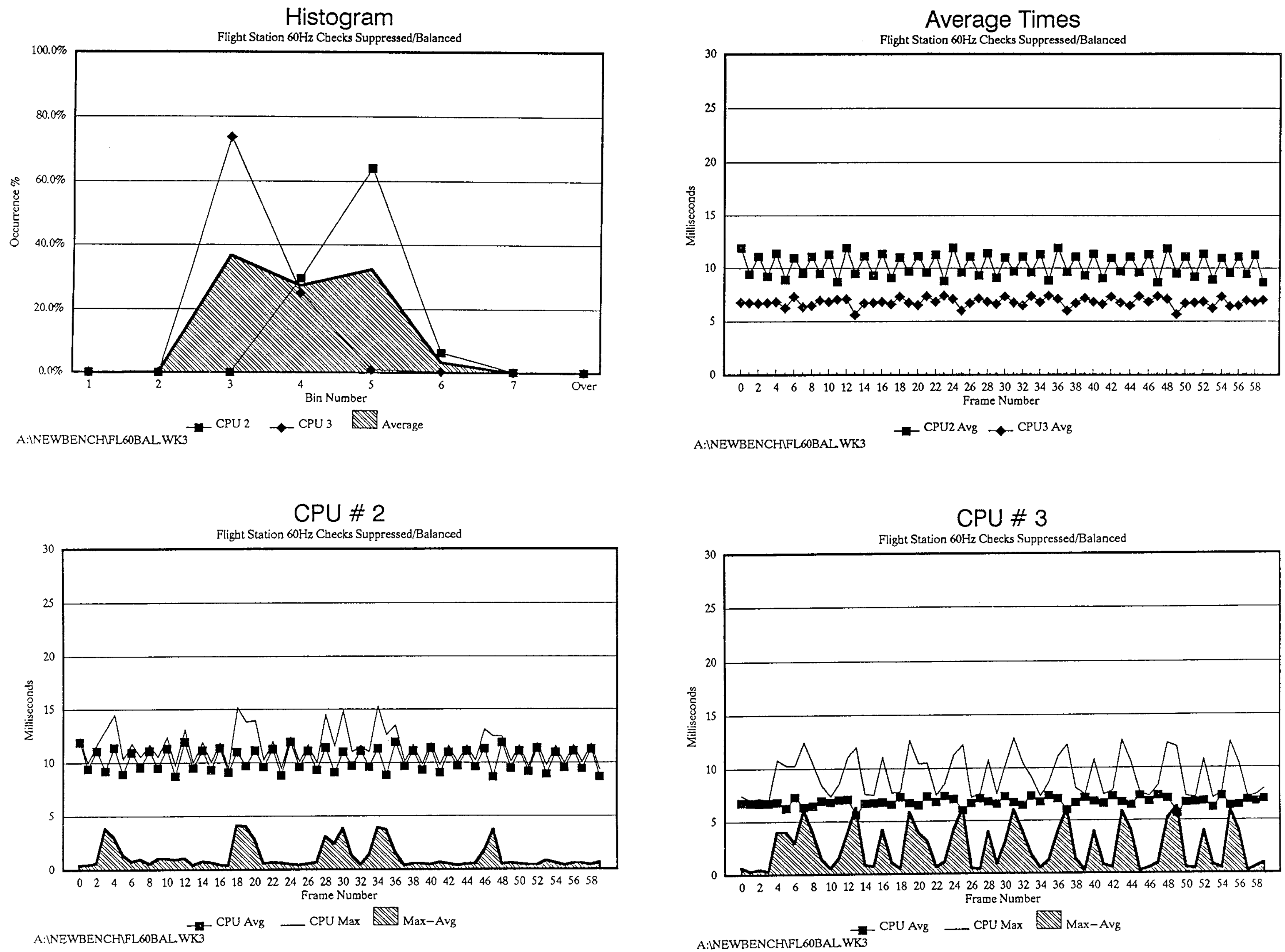
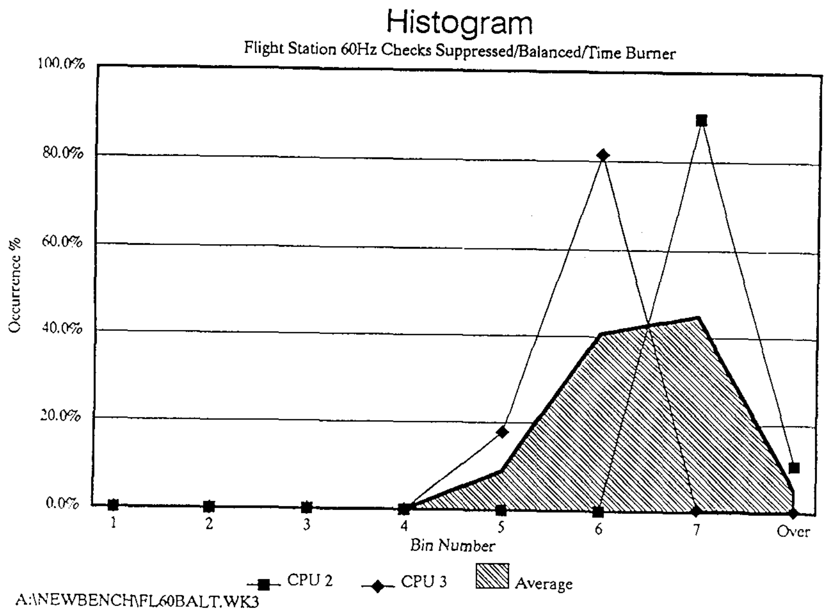
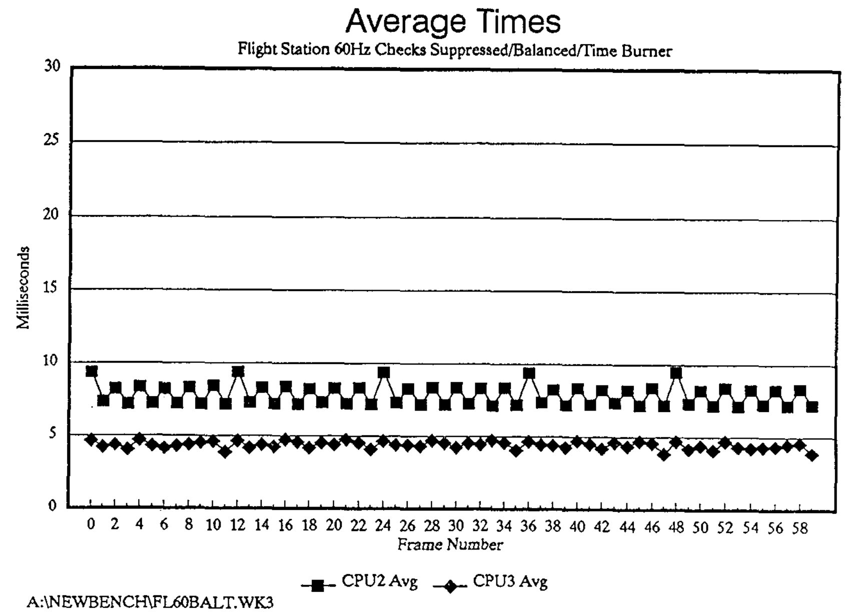
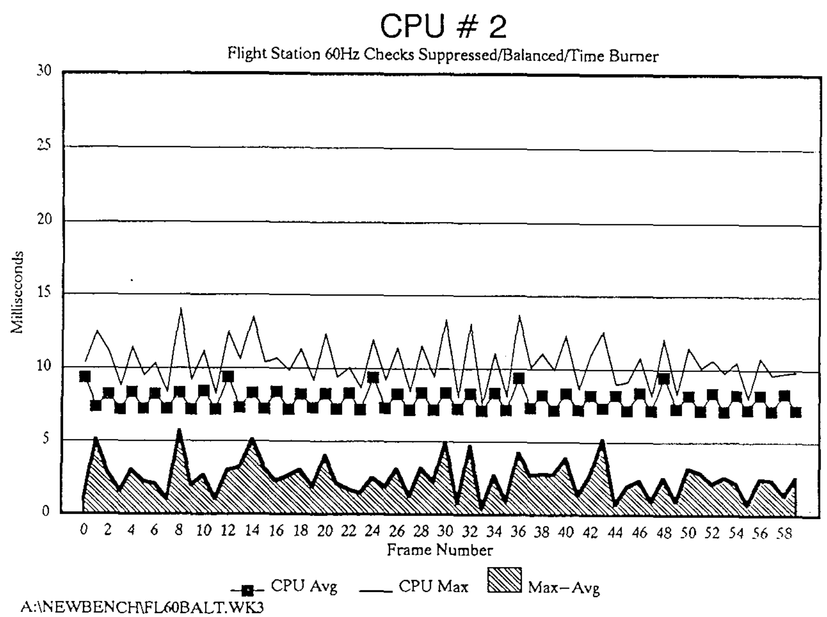


Figure 6 - Flight Station, 60 Hz, Checks Supressed, Load Balanced







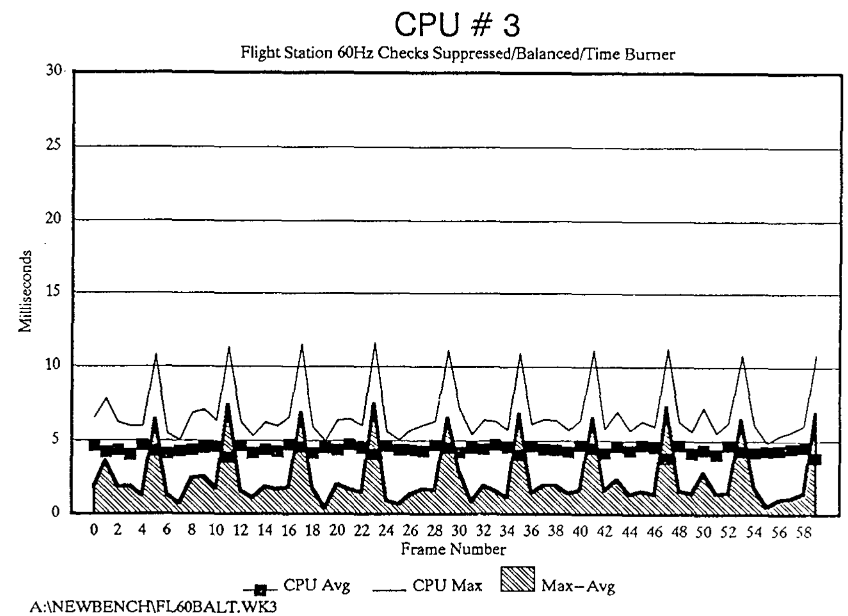
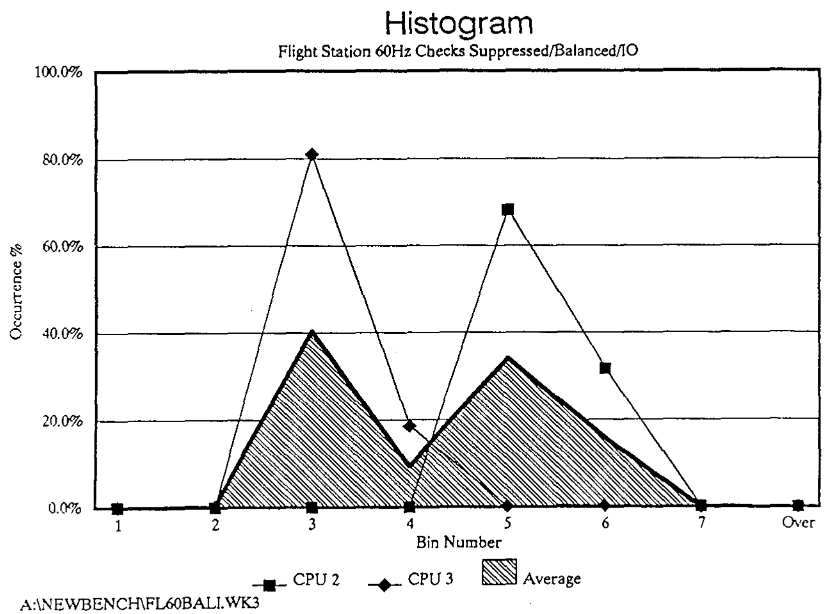
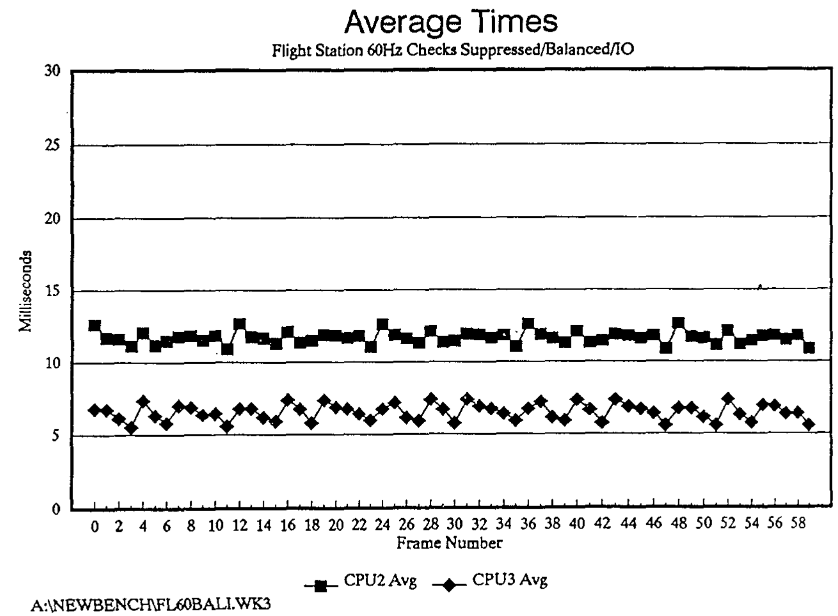
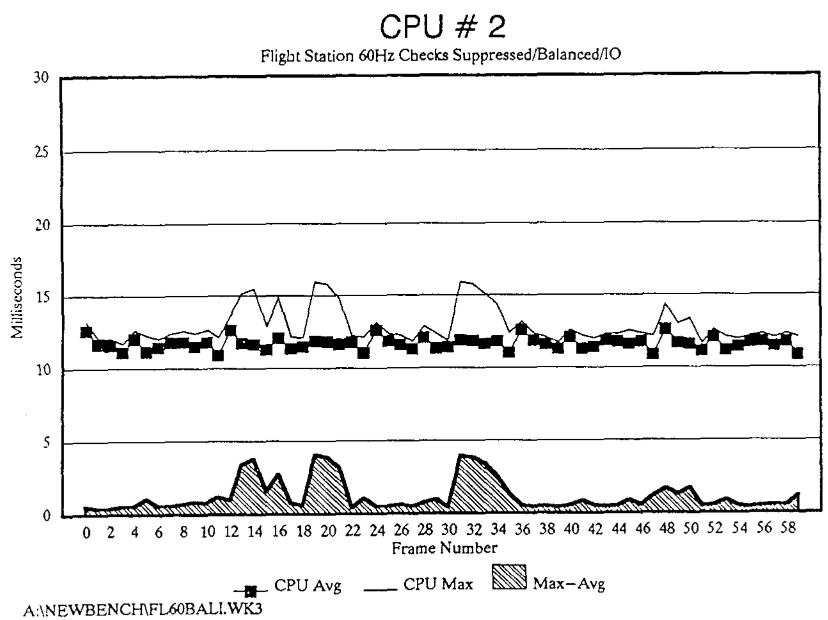


Figure 7 - Flight Station, 60 Hz, Checks Supressed, Load Balanced, 8 millisec Time Burner







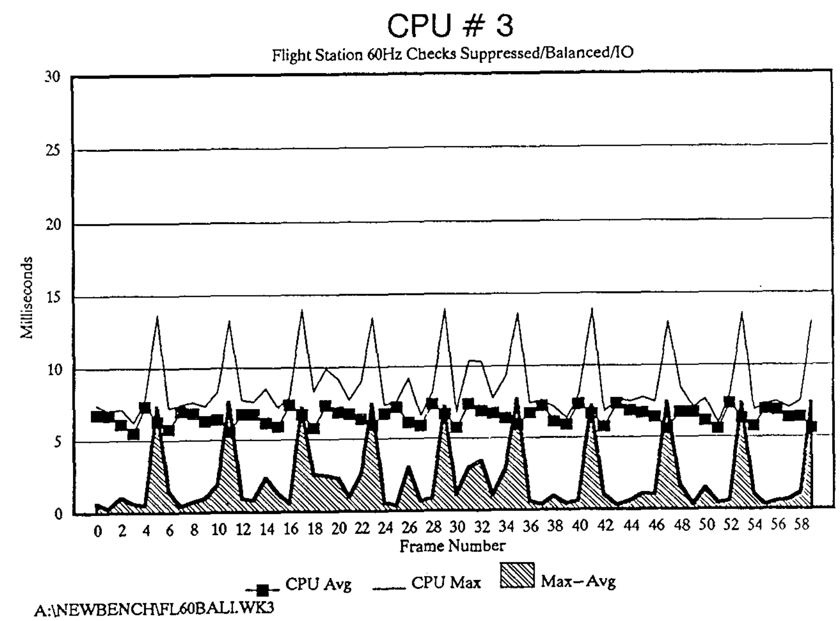


Figure 8 - Flight Station, 60 Hz, Checks Supressed, Load Balanced, HSD IO

PREDICTING NETWORK PERFORMANCE IN HETEROGENEOUS, MULTI-FIDELITY, SIMULATION NETWORKS

Christina Bouwens and Ron Matusof
CAE-Link Corporation
Binghamton, New York

ABSTRACT

Simulation networking is no longer new or novel. Heterogeneous, multi-fidelity networks have been successfully demonstrated using either proprietary protocols such as SIMNET, or Distributed Interactive Simulation (DIS) protocols. As the technology for simulation networking has matured, it has resolved some major issues. For example, we now have a standard for the exchange of information between networked simulations (IEEE-1278-1993). There has been very little work done toward prediction and accurate measurement of simulator network loading, and little significant work has been published concerning the implications of network loading toward the overall network fidelity and the successful transfer of training. Implicit in the underlying structure of the DIS is an assumption that network performance is purely an issue of applying appropriate technology to support a particular set of objectives. However, network loading imposes limitations upon these objectives and it is unclear what effect unexpected network performance has upon meeting a particular set of objectives.

This paper addresses the problem of predicting network loading in a heterogeneous, multi-fidelity simulation network. It discusses the issues associated with heterogeneous networks and multi-fidelity simulation. Using objective data obtained from a variety of networked exercises (both DIS and non-DIS) for context, this paper discusses the detailed issues involved in measuring network loading. Finally, it makes some recommendations for the future.

About the Authors

Christina Bouwens is a project engineer with CAE-Link Corporation, Binghamton, NY. She has over five years experience in simulator networking. She currently serves as systems engineer for several CAE-Link projects that use Distributed Interactive Simulation (DIS) and Aggregate Level Simulation Protocol (ALSP). Before joining CAE-Link, Ms Bouwens was the program engineer for the DIS standards program at the Institute for Simulation and Training. Ms. Bouwens has a Masters of Science in Mathematical Science from the University of Central Florida and a Bachelor of Science in Mathematics from Geneva College. She has published several papers on the subject of DIS and interoperability. Ms. Bouwens chairs the DIS working group for Communication Architecture and Security.

Ron Matusof is a staff engineer with CAE-Link Corporation, Binghamton, New York. He has over eleven years of experience in tactical, radar, and image simulation and in simulator networking. He has been an active participant in the DIS standards development process since the first DIS workshop in 1989, most notably in the area of fidelity definition. Mr. Matusof is currently the Principal Investigator for several CAE-Link research and development programs. He holds a Bachelor of Science in Electrical Engineering from the University of Pittsburgh. He has published several papers on the subjects of interoperability, mission rehearsal, image generation and cue correlation.

PREDICTING NETWORK PERFORMANCE IN HETEROGENEOUS, MULTI-FIDELITY, SIMULATION NETWORKS

Christina Bouwens and Ron Matusof
CAE-Link Corporation
Binghamton, New York

INTRODUCTION

Simulation networking is not a new concept and its use in large scale exercises has moved beyond proof-of-principle and into production. We now have a standard for the exchange of information between networked simulations (IEEE-1278-1993)[1], but the supporting infrastructure for implementation of a simulation network has not yet been completed. This leaves the designers of simulation networks with a myriad of questions concerning the implementation of a simulation network and little concrete methodology for predicting how the network will act under a variety of conditions.

There has been very little work published concerning prediction and accurate measurement of simulation network loading, and the implications of network loading toward network fidelity and successful transfer of training are not well understood.

Our overall objective is to develop methodologies for predicting simulation network performance and for determining its impact upon fidelity and transfer of training. This paper concerns itself only with the first part of the objective: prediction of network performance.

BACKGROUND

A simulation network is an arrangement which allows two or more simulations to communicate. A simulation network is a conceptual arrangement. It does not imply a particular type of communication media nor does it imply a set of communication protocols. These are implementations of a simulation network, and for a given network there are a large number of potential implementations.

Generally, simulation networks are governed by a network architecture. The architecture provides a set of design principles for the network implementation. Network implementations can be viewed from two perspectives -- the physical network and the virtual network. The physical network (Figure 1) describes the schematic and topological connection between network nodes, including the placement of nodes, the media through which nodes communicate, and the hardware/software which allow communication to occur. The virtual network (Figure 2) describes the logical interconnection between simulations, defined solely in terms of the flow of data and control.

The designers of simulation networks must be keenly aware of network performance. Network performance is the functional effectiveness of a network, and in the case of a simulation network, it is based on both the physical and virtual network implementations. The issues associated with measuring simulation network performance are derived from the network itself, the differences between individual simulation designs and differences in the level of realism across the network.

SIMULATION NETWORKING

Most simulation networks are a representation of a parallel processing methodology known as asynchronous data flow. Asynchronous data flow architectures are those in which the messages which flow between nodes provide control and synchronization of the system. In the data flow model, nodes are executed simultaneously, yet independently from each other. The output of the node depends only on the input and the function(s) that the node performs.

Simulation networks fit the asynchronous data flow paradigm in that each simulation is responsible for its own actions. Each simulation executes independently and simultaneously with other simulations on the network. All simulations are treated identically by the network because the functions of the simulation node depend only on the inputs provided by the network and the functions of the simulation.

Assessing the performance of network architectures is usually quite difficult due to the large number of parameters found in these systems. Because in asynchronous data flow architectures the nodes are functionally defined and are synchronized by the data

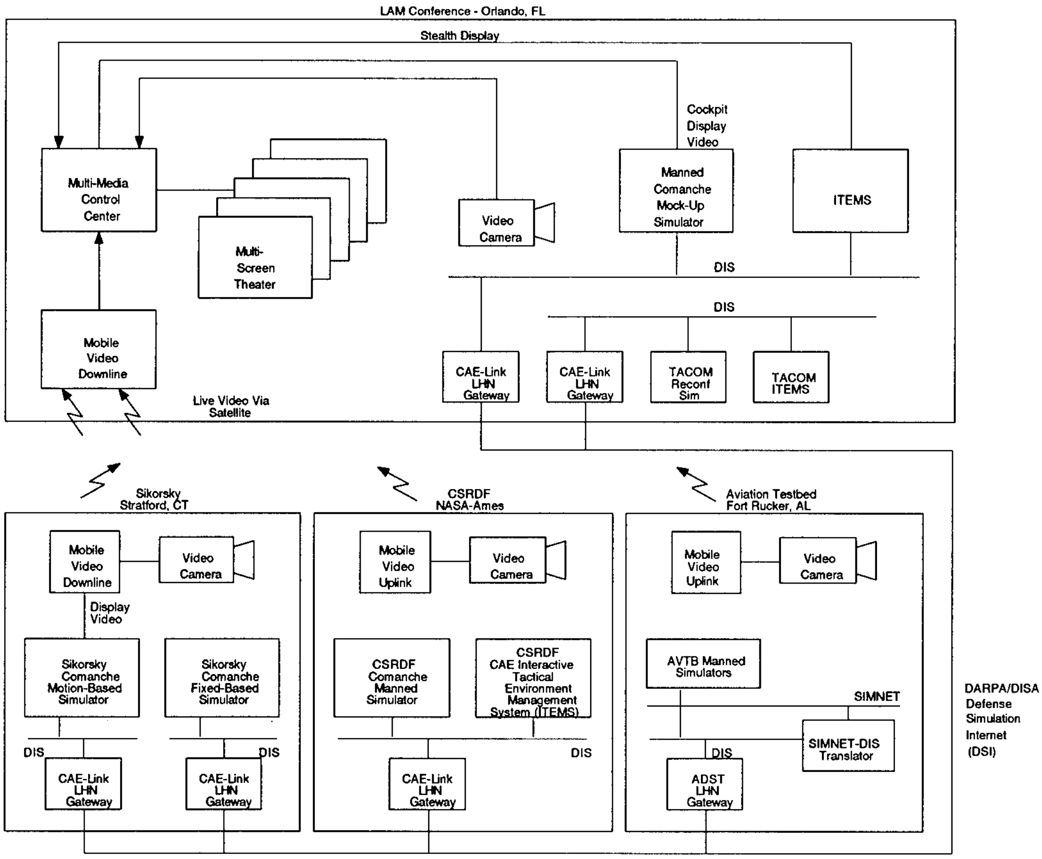


Figure 1
Physical Network

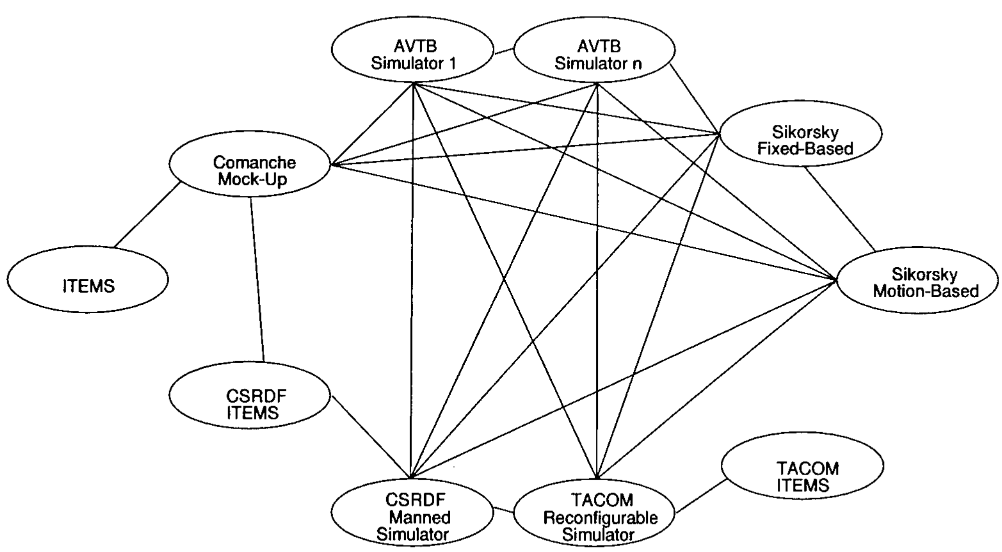


Figure 2
Virtual Network Example

that flow through them, meaningful prediction schemes can be developed which are independent of the characteristics of individual nodes. These schemes involve measurement and prediction of network loading based solely on the data flow. Schemes such as these can account for issues such as heterogeneous or multi-fidelity simulation.

HETEROGENEOUS SIMULATIONS

A simulation network is often characterized by the similarity between the individual simulations which comprise it. This characterization is associated with the physical network, and is usually divided into two domains: homogeneous and heterogeneous networks.

Homogeneous networks are composed of simulations which are essentially identical in design. Early distributed simulation networks, such as SIMNET, fall into this category. It is a relatively easy task to predict the network performance of homogeneous networks, since the interactions between two simulations on the network can be linearly extrapolated to almost any network size.

Heterogeneous networks, on the other hand, are composed of simulations of different design. Different vendors producing simulations for identical specifications will generally implement the simulations in different ways. For each simulation the implementation will probably be fully compliant with the specification, yet will likely vary greatly from other implementations.

DIS supports the networking of heterogeneous simulations. Although two heterogeneous simulations may identically meet a simulation specification for a particular non-networked application they may produce significantly different results in a networked environment. Because of this, it is more difficult to predict the network performance of heterogeneous simulations than for homogeneous simulations.

MULTI-FIDELITY SIMULATIONS

Compounding the problem of heterogeneous simulations is another problem concerning multi-fidelity simulations. A multi-fidelity simulation is one that has varying levels of fidelity depending upon its application. Fidelity is a characteristic of the virtual network and, in this case, is described as the degree of similarity between a simulation and the real world[2].

In simulation networking, multi-fidelity networks can be constructed where the simulations on the network are not necessarily of identical fidelity. This may occur because simulations on the network are designed to different specifications, they are designed to the same specification but implemented differently, or identical implementations of a specification or interfaced to the network in different ways and therefore behave with different levels of fidelity in the network environment (due to different filtering schemes, for example).

NETWORK PERFORMANCE

The designers of simulation networks will be expected to meet certain performance criteria for a particular simulation networking application. Unfortunately, the designer is left with almost no information as to how to predict network performance. Typical network performance criteria center around the physical constraints of the network, such as bandwidth and latency. While these are important criteria in determining overall network performance, they have little meaning without a corresponding set of virtual performance measures. There has been little research investigating the role of the virtual network in overall network performance.

To help determine the role of the virtual and physical network on overall performance, we reviewed data from five network exercises:

1. MULTISIM Experiments at Fort Rucker (1988):[3] This exercise involved the interconnection of four homogeneous, multi-fidelity devices via a proprietary (non-DIS) synchronous network transfer mechanism.

2. Project Desert STAARS (1991):[4] This exercise involved the development of a heterogeneous. multi-fidelity network of virtual and constructive simulations interconnected via a proprietary (non-DIS) synchronous transfer method.

3. I/ITSEC Demonstration 1 (1992):[5] This exercise was the first large-scale public demonstration of DIS, involving 18 manned and unmanned simulations, 22 listen only devices, and 1 live device. The network was multi-fidelity and heterogeneous and communicated using DIS 1.0 protocols.

4. I/ITSEC Demonstration 2 (1993):[6] [7] These exercises were a large-scale DIS exercise involving an increased number of manned and unmanned simulations, listen only devices, and live devices. Also included in this data is a SIMNET data stream from the Wright Flyer simulation of the DoD Dependent School demonstration. Again the network was multi-fidelity and heterogeneous and this time communicated using a slightly modified DIS 2.0.3 protocol.

5. CELLNET (1994): This was a small-scale exercise connecting a heterogeneous. multi-fidelity network of virtual and constructive simulations. These simulations were interconnected via DIS protocols implemented as an application layer transported via a synchronous transfer method.

Our goal in selecting these exercises was to allow us to study the effect on network performance as the construct of the network varies. We studied both homogeneous and heterogeneous networks for both DIS and non-DIS applications. Network transfer schemes varied and included both synchronous and asynchronous methods. In all cases, the networks were multi-fidelity.

We must point out that all five of these exercises were experimental applications of simulation networking and that there is no conclusive evidence for the validity of our observations. However, there were some very interesting trends which we observed.

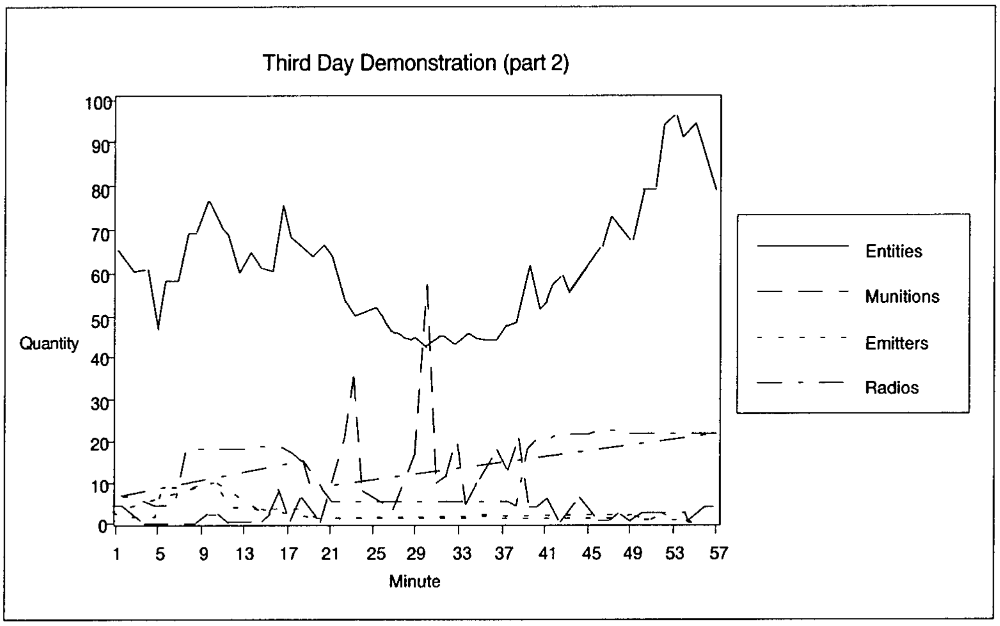
When evaluating network performance, the goal is to define how changes in either the virtual or physical network affects task performance The performance of the network is limited by the characteristics of both the physical and virtual network and by the mapping between the two. Bandwidth, latency, and throughput appeared to have the most pronounced impact on the performance of the physical network. Data synchronization and the interrelationship between network state updates (such as the issuance of PDU's) appeared to have the greatest impact on the performance of the virtual network. After reviewing our sample simulation networks, we noted the following trends:

Bandwidth: Bandwidth appeared to have no effect on network performance. Spare bandwidth ranged between 42% in the DIS flooding experiment during the 1993 I/ITSEC to 92% in the MULTISIM experiments. However, each of these exercises involved a small number of simulated entities. By definition, virtual networks have unlimited bandwidth. Therefore the design of a virtual network may be constrained by the bandwidth limits of the physical network. As a result, the designer of simulation networks must conciously determine if the virtual network can be appropriately mapped within the physical bandwidth limitations. Obviously, the mapping problem will get worse as the size of the virtual network (that is, the number of entities) grows.

Throughput: Throughput is the data capacity of a network. The throughput for the MULTISIM and Desert STAARS networks was almost constant over all applications, while the other networks exhibited "spikes" of activity of up to 35 kilobytes/second. These spikes appear to be related to the activity of entities on the network, and become significantly larger as the simulation workload increases. In the virtual network, data throughput is unlimited and spikes of activity pose no significant problem. However, the throughput of the physical network is constrained and these spikes affect the overall capabilities of the simulation network. In the data from the DIS exercises, the spikes increase in size and frequency when emission or radio PDUs are issued. Interestingly, there appears to be no correlation between the issuance of munitions PDUs and activity spikes (Figure 3).

Latency: The network designer, while concerned with the actual network latency, is more concerned with the effective latency of the network. Effective latency is the delay measured between an action initiated in one simulation and the action's representation by another simulation. It includes the latency of the physical network hardware as well as some additional delays introduced by the implementation of the network[8]. These delays include network transfer delay, network protocol delay, network transmission delay, network filtering delay, and network encryption delay.

We have limited empirical data concerning most latencies of the networks we studied. However, there is fairly good data concerning network transfer delay (the amount of time it takes to physically move data from a simulator to a network node). This delay ranged from 16 milliseconds to 200 milliseconds in the simulations for which it was measured. The total network transfer delay for a given interaction is the sum of the network transfer delay at the sending and receiving nodes. This means that in our sample networks, a maximum network transfer delay of 400 ms could occur.



	Entities	Munitions	Emitters	Radios
Averages	59.55932	6	2.101695	12.64407
Maximum	93	56	9	22
Minimum	41	0	0	4

Start of file: Wed Dec 1 20:00:00 1993
End of file: Wed Dec 1 20:59:59 1993
Total number of DIS PDUs 220717

Figure 3
Activity Spires vs. Issue of Munition PDU
(Courtesy Dr. Sandra Cheung[7])

There is insufficient data from the exercises that we reviewed to determine the effect of protocol delay (the delay introduced to a data stream due to the choice of network protocol) on overall network performance. However, subjective comparisons between the Desert STAARS and CELLNET networks (two networks which were different only in that one used a proprietary protocol while the other used DIS), revealed no discernible changes due to differences in protocol. Similar subjective observations[9] have been made between SIMNET and DIS applications.

Our limited data from the DIS exercises indicates that queuing delay (the delay which occurs as messages queue to be processed by a network node) had an insignificant impact upon network performance, provided that sufficient buffering exists at all network nodes. Networks appear to "deadlock" (message traffic ceases even though physical network is still active) when network receiving buffers overflow. More research is required before meaningful conclusions can be drawn.

Network filtering delay (the delay introduced by processing of asynchronous network state data updates) appears to become more pronounced as the allowable deviation between dead reckoned and actual entity state decreases. This is contrary to what we had expected, since with higher dead reckoning tolerances, more smoothing is required. Our theory is that a major cause of the filter delay is related to the rate at which entity state information is available from the network. When smaller tolerances are used, the dead reckoned position is corrected (and subsequently passed on the network) more often.[10] This implies that, on the average, there are more entity state PDUs to read. Since information is read from the network in a serial manner, an increase in the number of entity state PDU's implies that more time will be required to read and filter this information.

For example, the coordinate conversion processing measured at one node of the 1993 I/ITSEC demonstration was found to range from 30 μsec/entity when converting from geocentric to geodetic coordinates to 70 μsec/entity for the reverse transform.[11] There is, therefore, a 30 μsec/entity penalty to pay for each entity state PDU received. The more entity state PDUs are issued, the greater the filter delay. Interestingly, the coordinate conversion processing times were found to vary based on the desired accuracy of the conversion (a 0.004 foot error required four iterations of the conversion).

We believe that a similar effect will happen for encryption and transmission delays, but there is insufficient data in the exercises that we studied to either support or disprove this contention.

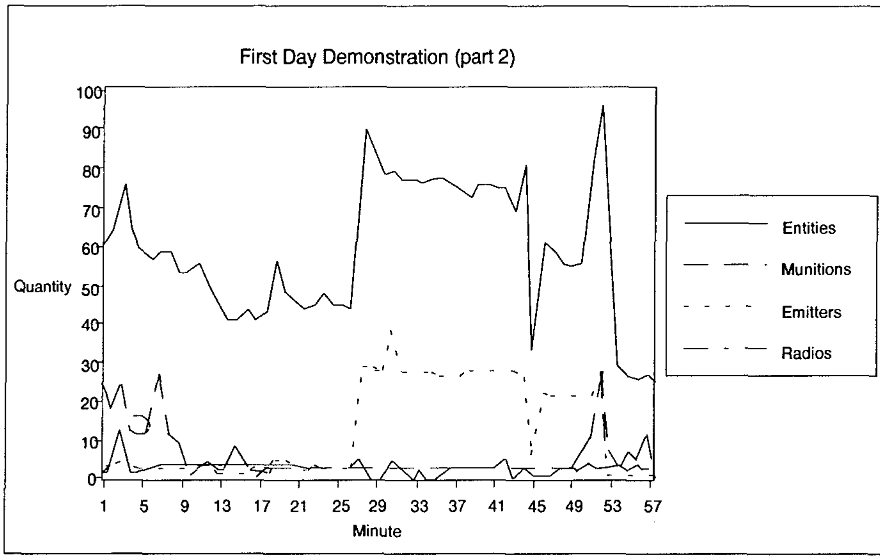
Delay dispersion: Unlike the cues in a single simulation, the latency of individual cues on the network has a component which is both random and unbounded. Therefore, not only is the delay of an individual cue important, but the variation in the length of the delay (an effect known as delay dispersion) is important as well. The variance in delay can cause a disordering of packets such that the sequencing of state information (player position and velocity, for example) is incorrect. We observed no dispersion in the MULTISIM and Desert STAARS synchronous networks and a very small amount of delay dispersion in the other networks. There is no evidence that packet disordering caused any network performance problems.

Data Synchronization: The simulation network represents an asynchronous data flow architecture. Asynchronous data flow architectures are, by their nature, synchronized by the flow of data between nodes. In this architecture, data is synchronized only at the network nodes. The simulations themselves are not synchronized by the data flow. Therefore, we often observe data synchronization problems in all asynchronous applications (including CELLNET, where DIS was applied as an asynchronous application layer over a synchronous transfer method). In synchronous virtual networks, data flow is, by definition, time coherent. When coherence was lost (due to loss of synchronization signals, for example), we observed catastrophic failure in that the system could not automatically resynchronize and erroneous data was produced. The skewing of data and the inability to resynchronize it caused several cases of "extrapolation induced oscillation." When this occurs, extrapolations based on erroneous data produce increasingly inaccurate results until the extrapolations themselves become unstable and the simulation becomes unusable.

Interrelationship of PDUs: Certain information fields, such as position and attitude, are repeated in several different PDU types. The assumption is that an antenna, which is generally offset from the center of a vehicle, may move out of a positional tolerance without the vehicle moving at all. For this case, we would need to perform a tolerance check on the position and attitude of the antenna, and issue new PDUs whenever the antenna goes out of tolerance. In the DIS applications that we studied, and in particular the 1993 I/ITSEC demonstration and the CELLNET exercise, we observed a stunning interrelationship between Entity State PDU generation, Emission PDU generation, and Radio Emission PDU generation. Increases in the generation of either Emission or Radio PDUs resulted in a two-fold increase in the generation of entity state PDUs (Figure 4). This was highly unexpected, but can be observed in all 1993 I/ITSEC DIS demonstrations and in recorded data from the CELLNET exercise. The implication of this trend is that the use of emission or radio PDUs may affect network performance in a disproportionate manner than other types of PDUs. We believe that more information must be gathered before this trend can be considered more than coincidental.

Dead Reckoning Thresholds: Dead reckoning thresholds directly affect the amount of entity state traffic on a DIS network. It has been shown that network traffic can be reduced by up to eighty per cent by using a dead reckoning algorithm.[12] However, this reduction in network traffic was accomplished by allowing vehicle appearance to vary up to three degrees in rotation and up to ten per cent of the vehicle's dimensions in position before a state update is required. In all of the DIS applications that we studied, the threshold was always set to 1 meter and 3 degrees. Therefore, we have no data to determine the impact of varying the thresholds in these exercises.

Non-Simulation Network Traffic Non simulation traffic appeared to be a problem in the I/ITSEC demonstrations. It is reasonable to assume that simulation networks will not, in general, occur on pristine networks. Therefore, the non-simulation network traffic must be quantified prior to any prediction. Again, we were unable to quantify the effect of non-simulation traffic on the performance of any of the exercises that we studied.



Averages	53.79661	4.644068	11.38983	2.4068
Maximum	88	25	35	11
Minimum	22	0	1	1
StDev	16.29341	6.180511	11.08745	1.274669
Totals	151	73	40	15

Start of file: Tue Nov 30 11:00:00 1993
End of file: Tue Nov 30 11:59:59 1993
Total number of DIS PDUs 266055

Figure 4
Issue Rate of Emission vs. Entity State PDUs
(Courtesy Dr. Sandra Cheung[7])

MEASURING NETWORK PERFORMANCE

Network performance is a complicated combination of a number of factors. We have tried to define a reasonable measurement paradigm which allows the network designer to determine the feasibility of a network design prior to its implementation. Our methodology relies on determining the available bandwidth and effective latency of the network, and from these we determine the maximum number of entities that can be supported by the network given the exercise requirements that the network must support.

We have based part of this analysis on work presented at the 1993 I/ITSEC.[13] In this work, a four step program was outlined to estimate bandwidth requirements:

1. Document assumptions about minimum attributes of each entity class represented
2. Estimate the exercise bandwidth requirement to approximate actual PDU issue rates
3. Determine the number of entities (and tactical links) required for an exercise
4. Calculate exercise bandwidth based on these individual estimates.

One problem with this methodology is that it relies heavily on an exercise designer's ability to estimate exercise requirements. Additionally, this does not address the performance issues introduced by mapping the virtual network onto the physical network. Our hope was to develop a similar methodology which does not rely on the exercise designer's a priori knowledge of networking.

Our methodology involves determining a set of equations which can be used to determine the worst case latency and bandwidth of the physical network, applying knowledge of the intended exercise and the simulations involved to determine worst case PDU issue rate (a characteristic of the virtual network). We then combine these performance measurements to determine the maximum number of entities the network can support under worst case conditions. In other words, we map the virtual network onto physical network, and then bound the network performance by the physical constraints.

The latency of the system is heavily dependent upon applications and upon the transmission medium and protocol selected. For any individual path, the worst case latency is the sum of the individual delays in the system, namely the transfer delay (D_t), the protocol delay (D_p), the queuing delay (D_q), the filtering delay (D_f), the transmission delay (D_x), the encryption delay (D_e), and the worst case delay dispersion (D_{disp}). The first four factors are multiplied by two to account for delays at both the transmitting and receiving nodes (Figure 5).

$$L_{wc} = 2\,(D_t + D_p + D_q + D_f) + D_x + D_e + D_{disp}$$

The maximum bandwidth of any network is a known quantity. It is physically impossible for a network of simulations to exceed this bandwidth. The effective bandwidth available to a simulation node is determined by taking the maximum bandwidth and subtracting out the overhead due to protocol, and the bandwidth used for non-simulation network traffic.

$$BW_{eff} = BW_{max} - BW_{overhead} - BW_{other}$$

Bandwidth may affect the message delay time at network nodes in cases where variable intensity traffic exists (such as most DIS exercises). In these cases, the network may be modeled as a Poisson message data stream, and the effects of limiting bandwidth on the network queues can be predicted in a relatively straightforward manner.[14] One can then predict if a exercise will meet a particular network requirement. For the networks we reviewed, bandwidth had no effect on latency (or vice versa), since the worst case loading still had 42 per cent spare capacity.

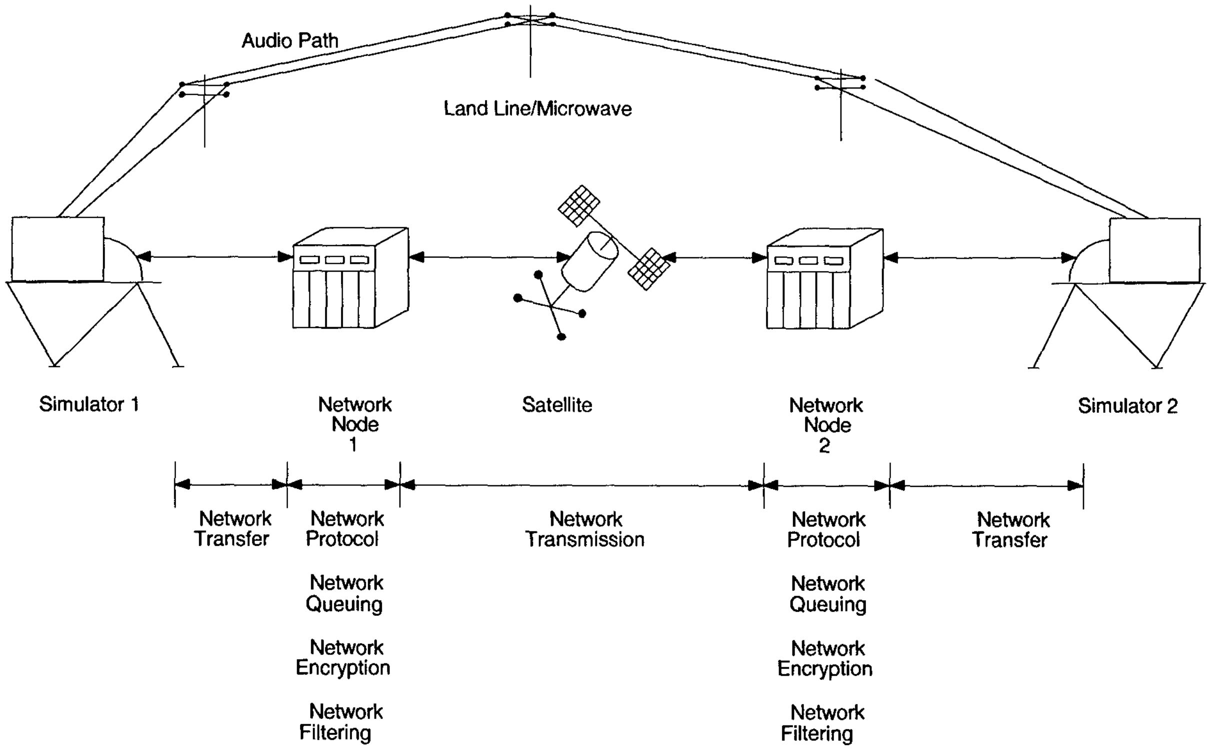


Figure 5
Typical Time Delays for Long-Distance Networked Systems

In order to determine the maximum number of entities which a network can support, we must first determine the worst case rate at which PDUs will be issued. This is done on an entity by entity basis In the networks that we studied, six PDU's (entity state, detonation, emission, transmitter, signal, and laser) make up the bulk of the message traffic. Based on an analysis of the data we studied, we found that this traffic accounted for an average of 98.3 per cent of the network traffic. On the average, PDU size was 1384 bits. Using the issue rates obtained from our sample exercises, we determined that a worst case average of 200 bits per second is required for all other PDUs. We can therefore aggregate all of these other PDUs into one representative PDU with an issue rate of 200 bits/second. We intend to adjust this aggregate representation as we analyze more data. Using the formulae published in the proposed IEEE Standard Draft version 2.0.3,[15] we can get a rough idea of the predicted load. These formulae are similar to those derived by Doris and Loper[16] in 1993, but include DIS 2.0.3 Draft Protocols. Our intent was to refine these equations to also take into account the worst case rate of issue, R, of the PDUs.

R is determined by looking at the threshold values set for the dead reckoning algorithm, an entity's capabilities and the entity's dimensions. PDUs are issued whenever the difference between an entity's dead reckoned position (P_{dr}) and its actual position (P_{act}) exceeds the positional tolerance (T_p). The tolerance can be represented in terms of the entity's velocity by substituting velocity–time products in place of instantaneous positions:

$$T_p = |P_{dr} - P_{act}| = \Delta t(V_0 - V_{n-1})/\text{frame rate}$$

where V_0 is the velocity at the time that the last entity state PDU was updated for a given entity, V_{n-1} is the velocity calculated by the last pass of the real–time simulation of the entity, and the frame rate is the iteration rate of the simulation. The worst case occurs at maximum entity velocity (V_{max}). Solving for Δt and substituting V_{max} for V_0 yields:

$$\Delta t = 1/V_{max}\,(T_p + V_{n-1}/\text{frame})$$

In the worst case, tolerance is simultaneously broken in both position and orientation ($T\omega$). In this case, the maximum value of V_{n-1} is the projection of the dead reckoned velocity onto the actual velocity vector, where ω is the angle between the two vectors. Substituting the projected V_{max} for V_{n-1} we get:

$$\Delta t = T_p/V_{max} + \sin(T\omega)/\text{frame} \quad \text{and} \quad R = 1/\Delta t$$

Our table of formulae, then, is shown in Figure 6:

PDU	FORMULA for size estimate	where:
Entity State	$R(1152 + 128A)$	A = # of articulated parts
Detonation	$800 + 128H$	H = # of articulated parts hit
Emission	$R(192 + E(160+ B(416+64T)))$	E = # of emitters
Transmitter	$R(768 + S_m M)$	B = # of beams per emitter
Signal	$256 + L$	T = # of targets in beam
Laser	576	M= # of Modulation parameters
Other PDUs	200	S_m = Size of modulation pattern m
		L = Length of data stream

Figure 6 PDU Sizing Formulae

We next sum all PDU issue rates (in bits/second) over all of our entities in order to determine the loading of the network:

$$\text{Virtual Load} = \sum_{\text{entities}} \text{PDU bits/second}$$

Finally, we determine the maximum number of "average" entities which can be supported by this network.

$$\text{Max Entities} = BW_{eff}/\text{Virtual Load}$$

This number can be used for planning purposes. It represents an average worst case for the network, given the physical constraints of the network and the exercise goals. The network designer can now assess alternatives and their impact on the network's physical and virtual design. For example, the designer may choose to improve the accuracy of geo–positioning., but does so at the expense of increased latency.

The formula for max entities is reciprocal and can be used to derive the required bandwidth given a desired number of entities with known capabilities:

$$BW_{eff} = \text{Max Entities} * \text{Virtual Load}$$

CONCLUSIONS

We have developed a method for predicting the maximum number of entities that can play in a network exercise given the constraints of the physical network, the exercise objectives, the characteristics of the virtual network, and latency requirements of the exercise. The observations and derivations which we have made in this paper are based solely upon five experimental exercises conducted over the last several years and published research.

A great deal of research remains to be conducted in this area. Our data set was limited and the data was not collected with the expressed purpose of developing prediction methodologies for network performance. Future experimentation concerning the effects of bandwidth, latency, delay dispersion, data synchronization, interrelationship between PDUs and implications of bandwidth reduction methods (such as dead reckoning) remain to be conducted. Specific future areas of research concerning network performance include:

- Protocol Delay
- Queuing Delay
- Encryption Delay
- Transmission Delay
- Non-simulation Network Traffic
- Interrelationship between PDU types
- Impact of varying latency
- Analysis of operational systems (as opposed to experimental or demonstration)
- Comparison between live, virtual, and constructive simulations.
- Impact of network performance on fidelity
- Impact of performance on transfer of training

As a further step in this study, we intend to apply our methodology toward prediction of upcoming DIS exercises (including the 1994 I/ITSEC demonstration) and fine tune this algorithm as needed.

Prediction of network performance will provide only part of the information that the network designer must know prior to building a simulation network. Even the best prediction algorithm will not allow the designer to develop successful network exercises unless it is accompanied by knowledge of how the network implementation affects transfer to the real-world. When equipped within this knowledge, simulation networking can reach its true potential.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the invaluable assistance of Dr. Sandra Cheung from the University of Central Florida Institute for Simulation and Training and Dr. Joshua Seeger from BBN Systems and Technologies. The authors also acknowledge the efforts of STRICOM in supporting DIS research initiatives.

REFERENCES

[1]IEEE Standard 1278-1993, IEEE Standard for Information Technology -- Protocols for Distributed Interactive Simulation Applications Entity Information and Interaction, IEEE Computer Society, May, 1993.

[2]Hays, R. T., "Fidelity: A Concept Paper", Army Technical Institute Technical Report 490, Alexandria, VA, November, 1990.

[3]Dees, J.W. and Cornett, T.R., "Simulator Networking in Helicopter Air-to-Air Combat Training", AIAA Flight Simulation Technologies Conference, Boston, MA, September, 1989.

[4]Monette, R., Knight, S.K., and Goodwin, P., "Sustainment Training for Army Aviation Readiness through Simulation", Interservice/Industry Training Systems Conference, Orlando, FL, December, 1991.

[5] Loper, M., Goldiez, B., and Smith, S., "The 1992 I/ITSEC Distributed Interactive Simulation Interoperability Demonstration", Interservice/Industry Training Systems and Education Conference, Orlando, FL, December, 1993.

[6]Seeger, J., "Networked Oriented Scalability", Proceedings of the 10th Workshop for the Interoperability of Defense Simulations, Orlando, FL, March, 1994.

[7]Cheung, S.E., "I/ITSEC 1993 DIS Interoperability Demonstration Traffic Analysis", Proceedings of the 10th Workshop for the Interoperability of Defense Simulations, Orlando, FL, March, 1994.

[8]Sawler,R.J., and Matusof, R., "Issues Concerning Cue Correlation and Synchronization of Networked Simulators", AIAA Flight Simulation Technologies Conference, New Orleans, LA, August, 1991.

[9]Panzitta, M.J., and Moore, R.G., "Visual System Interoperability Between CCTT and SIMNET", IMAGE VII, Tucson, AZ, June 12-17, 1994.

[10]Economy, R., Ferfuson, R., and Pollak, E., "Geographic Position Accuracy in a Distributed Simulation Environment", IMAGE VII, Tucson, AZ, June 12-17, 1994.

[11]Ibid.

[12]Miller, D., Pope, A., Waters, R., "Long-Haul Networking of Simulators", 10th Interservice/Industry Training Systems Conference, Orlando, FL, December, 1988.

[13]Doris, K. and Loper, M., "DIS Network Traffic Analysis Estimation Techniques", Interservice/Industry Training System and Education Conference, Orlando, FL, December, 1993.

[14]Golovko, N.I., "Message Delay Time at Network Node for the Case of Incoming Traffic of Variable Intensity", Avtomatika i Vychislitel'naya Tekhnika, Volume 23, Number 2, 1989.

[15]Proposed IEEE Standard Draft, IEEE Standard for Information Technology -- Protocols for Distributed Interactive Simulation Applications Version 2.0, Third Draft, IST CR-93-15, Institute for Simulation and Training, May, 1993.

[16]Doris, K. and Loper, M., "DIS Network Traffic Analysis Estimation Techniques", Interservice/Industry Training System and Education Conference, Orlando, FL, December, 1993.

SIMULATION AND TRAINING SYSTEMS SUBCOMMITTEE

Chair
Kathy Walsh, SAIC

Deputy Chair
Ron Sheffer, Loral Federal Systems

Members
Bill Simmons, ECC
Bill Walsh, MEI Technology Corporation
Bob Epps, CAE-Link Corporation
Col Michael Whittenberg, DAMO-TRS
Don Jacobs, PULAU Electronics Corp.
Frank Lawler, Syntech
Gary Lewis, Loral
Jerry McLemoore, Naval Air Maintenance Traininng Group
Jim Grebey, Contraves
LCDR Jim Hooper, NAWCTSD
Maj. Chuck Delair, MARCORSYSCOM
Mark Adducchio, USAF
Richard Gilmour, STRICOM
Scott Royse, Southwest Research Institute
Steve Eaton, Chief of Naval Air Training
Tom Lockhart, USAF

Section 5
Table of Contents
Simulation and Training Systems Papers

INNOVATIVE SONAR TRAINING DESIGN: LINKING SONAR CONCEPTS WITH FAMILIAR HUMAN CONCEPTS

Dr. Thomas J. Hammell
Mr. Frederick M. Ewalt
Paradigm Associates
East Lyme, CT

Dr. Robert Ahlers
Ms. Cathy Matthews
Naval Air Warfare Center Training Systems Division
Orlando, FL

ABSTRACT. An instructional approach was developed for training applied sonar skills. The approach allows a student to effectively apply concepts learned in a classroom to problems presented on a training simulator. Instruction is in the form of visualizations integrated with the training simulation. These visualizations of environmental, tactical, and acoustic variables facilitate training by providing links from simulation elements to their more abstract representations on the tactical console. Information which addresses the procedural aspects of operating the tactical console is included in the training approach. This approach was presented to submarine sonar instructors and students, as a series of static display snap-shots in the context of specific training scenarios. Evaluation was based on their judgments, obtained with a structured-interview questionnaire, addressing the overall instructional approach and prototype display/control design features. The value of this type of instructional assistance was found to be very high.

Dr. Thomas J. Hammell is President and Chief Scientist of Paradigm Associates. He has directed programs for Navy, Maritime Administration, Coast Guard, and private industry; addressing simulator training systems, advanced instructional concepts, human-computer interface, command and control decision making, port and waterway development, staffing standards, aids to navigation, naval tactics, and certification of mariner training and licensing systems. He has been a faculty member in Computer Science and Psychology at the University of Connecticut. He received Bachelor of Engineering and Master of Management Science degrees from Stevens Institute of Technology, and a Ph.D. in experimental psychology from the University of Connecticut.

Mr. Frederick M. Ewalt is a Senior Program Engineer with Paradigm Associates. He has had over forty-two years of combined military/industry experience in anti-submarine warfare combat systems, including over thirty-four hundred hours of formal USN service schools and technical courses. He retired from the Navy at the rank of LCDR, after twenty-three years of enlisted and commissioned submarine duty. His expertise covers submarine systems, focused in sonar and tactical areas. His recent interests include simulation modeling, mission task analyses, definition of display requirements and designs, and identification of operational and training requirements.

Dr. Robert Ahlers is a Research Psychologist with the Human Systems Integration Division of the Naval Air Warfare Center Training Systems Division. He has managed research projects concerned with the application of knowledge-based modeling to the simulation of intelligent agents within a training environment. His recent work addresses the use of artificial intelligence techniques to provide certain instructional features, such as performance diagnosis and feedback, sequencing of instructional materials, and tactical adversary capabilities, within an embedded training environment. He graduated from the University of Virginia with B.A. and M.A. degrees in experimental psychology and from North Carolina State with a Ph.D. in Human Factors.

Ms. Cathy Matthews is an Electronics Engineer with the Advanced Simulation Concepts Division of the Naval Air Warfare Center Training Systems Division. She has worked in the ASW Acoustics area on submarine, surface, and air community trainer procurements. She recently began working in the research area investigating new approaches to training devices in support of greater training feedback and lower cost constraints. She graduated from Florida Atlantic University with a B.S. in Ocean Engineering, and from the University of Central Florida with an M.S. in Computer Engineering.

INNOVATIVE SONAR TRAINING DESIGN: LINKING SONAR CONCEPTS WITH FAMILIAR HUMAN CONCEPTS

Dr. Thomas J. Hammell
Mr. Frederick M. Ewalt
Paradigm Associates
East Lyme, CT

Dr. Robert Ahlers
Ms. Cathy Matthews
Naval Air Warfare Center -Training Systems Division
Orlando, FL

INTRODUCTION

Traditional DoD training strategies have focused on face validity to maximize the transfer of training, based on research that has shown such designs as often positively related to transfer (Osgood, 1949; Roscoe, 1980). Recent research has shown significant instructional enhancement results from use of strategies which present information corresponding to student internal cognitive representations, as summarized by Maxey, Scopatz, Madden & Ahlers (1993). These findings foster creative instructional media designs which key on fundamental human cognitive thought structures to provide a link between familiar concepts and difficult-to-comprehend operational system characteristics. The findings suggest that applied training strategies, such as for sonar employment training, should include media with good operational face validity supplemented by instructional information familiar to student cognitive representations.

Effective employment of modern submarine sonar systems is exceptionally complex. The sonar employment process is confronted by increasing equipment sophistication, quiet adversaries, changing roles of the submarine force, and ever-changing tactics and operational techniques. This complexity is further compounded by submarine operations' growing reliance on environmentally dependent information. The modern sonar system, including its operators, is continually challenged to extract increasing amounts and quality of information from a complicated environment. The concepts of sonar operation and the environment are often obstruse, with sophisticated and involved operating procedures. Achievement of the requisite operating precision requires exceptional levels of operator proficiency to compliment the advanced sensing and processing equipment. Effective operator skills and knowledge require the development of sophisticated, and often abstract, cognitive structures pertaining to the acoustic medium and operational system. It is postulated that the process of learning these will benefit from instructional strategies that augment the operational system displays with information enhancing student understanding and visualization of sonar concepts.

The training process could link new sonar concepts with already-developed familiar student cognitive representations, and then with operational sonar system displays and controls. This congruence, together with imaginative information presentation and visualization approaches, is expected to greatly enhance the student's grasping of fundamental sonar concepts and their application.

To assist in student acquisition of robust cognitive representations of sonar employment concepts, an instructional approach was developed to provide the student with alternative views of task-relevant sonar information, emphasizing visualization imagery, to augment information normally available from the operational equipment. The alternative view information, presented on a student-controlled instructional display placed alongside the operational sonar console trainer, would present a stylistic characterization of sonar system physical phenomena in graphical formats familiar to the student. This linkage of relevant applied information in the operational format with similar information in a more cognitively-familiar format is postulated to assist in student learning and understanding of sonar concepts and their applications, building cognitive representations and strengthening their association with the operational sonar equipment/information.

BACKGROUND

The broad learning/training research literature provides a wealth of guidance to support the design of the postulated alternative view instructional strategy in an applied setting. Certain issues of direct relevance to the postulated strategy are identified below.

Student Cognitive Representations

A common theme is emerging from the literature of perception, decision-making, expert vs. novice problem solving, performance under stress, team functioning, and training -- centered on the importance of well-developed cognitive structures and representations to human performance (Hammell and Ewalt, 1994). For example, cognitive representations have been described as fundamental to human decision making (e.g., Klein, 1989), and expert performance (e.g., Means, Salas, Crandall, and Jacobs, 1993); shared mental models have been identified as conducive to improved team performance (e.g., Cannon-Bowers, Salas, and Grossman, 1993). Individuals are believed to posses internal cognitive representations of objects, events and time-sequenced processes (e.g., Weiten, 1992) that are fundamental to human thinking and thought processes. These representations can have various forms, such as a mental image of an object (e.g., towed array conical beam pattern); or a mental image of the TMA process to calculate target range and course, in the form of a template; or a mental image of a time-line sequence of events for resolving sonar bearing ambiguity, in the form of a script. Individuals are believed to have many overlapping representations, learned from and corresponding to their many and varied experiences through life.

Visual Imagery

One of the primary goals of sonar operator employment training is to provide an understanding of the complex relationships among acoustic signals, the underwater environment, and the acoustic processing equipment. This understanding is necessary to allow an operator to effectively deal with novel tactical situations. Robust, appropriate cognitive representations are necessary to enable an operator to most accurately perceive the sonar-tactical situation as it exists, and to conduct effective decision making and problem solving activities. Since the perceptual process is dependent on past experience, the training strategy should start with, and build on, a student's existing cognitive representations to achieve an efficient learning process. That is, sonar training should begin with cognitive representations of familiar experiences and build to those of the sonar system and its employment. Furthermore, the training system should provide substantial instructional information in a visual format, since visual imagery has a beneficial effect on memory and information processing of complex information (e.g., Pavio, Smythe and Yuille, 1968). For example, representation of sonar beams (an abstract quantity not seen) can be accomplished in a visual manner that can be easily seen and understood by the student. A 3-D beam image, similar to a light beam, is an example of an initial familiar context that can be expanded into appropriate sonar beam mental images. To assist in forming a strong cognitive representation, the student would be able to rotate his viewpoint of the own ship-target-beam imagery in the 3-dimensional environment to provide a better understanding and visualization of the complex beam characteristics. Information in this abstract visualization should in-turn be linked with pertinent information available on the operational sonar console.

Instructional Principles

The research literature is replete with findings pertinent to the design of instructional strategies for applied training applications. Maxey et al (1993) provides a summary of many relevant principles, such as learner centered control of instructional information (i.e., provision of assistance only when requested by the student). Examples of principles considered to have direct relevance to development of cognitive representations for applied sonar operator training, during hands-on exercises using operational-like sonar consoles, are:

Instructional process:

- Student free-exploration, within a guidance structure;
- Procedural and non-procedural tasks supported with aiding information;
- Self-paced, student-selected assistance;
- Aiding information to student to reduce errors during early stages of basic training;
- Reduced aiding during later stages of training, or at any time.
- Positive guidance when student requests help;
- Immediate and delayed reinforcement, tailored to student progress;

Instructional Information:

- Augment operational displays;
- Address fundamental relationships and operational processes;
- Functional information content, specific to operational function and task;
- Hierarchical information organization, essential/minimal information at each level;
- Link with prior classroom instruction and operational manuals;
- Provide alternative views of problem situation;
- Visual/graphical information emphasis, using textual information as necessary.

These were incorporated into the instructional strategy design process under the alternative view approach.

Objective

The objective of the study was to design and evaluate an instructional approach for sonar employment training that provides alternative information to the student, during hands-on sonar trainer exercises, augmenting the information available on the trainer's operational sonar console. The information presentation emphasis was to accentuate the use of visualization imagery, to better-assist the student in acquiring robust cognitive representations.

METHODOLOGY

The research was conducted in the context of applied training for the AN/BQQ-5 submarine sonar system. Difficult-to-comprehend sonar concepts were identified with the assistance of the staff at the U.S. Naval Submarine School. Instructional aiding display concepts were developed to assist in teaching the understanding of two sonar procedures: 1) towed array bearing ambiguity resolution, and 2) relative motion/target motion analysis (TMA). The display concepts were developed to present information in formats believed to correspond with already-developed student internal cognitive representations (i.e., present complex sonar information in a visual manner that would be easily recognizable by the students). The design approach combined applied instructional principles with computer-based visualization techniques, to address applied problems.

The potential value of the alternative view approach was assessed at the Submarine School based on judgments of eight sonar instructors and students, all of whom were experienced sonar operators. The alternative view instructional features were illustrated in the context of two applied sonar training scenarios, one each for the towed array and relative motion/TMA display designs. The scenarios were presented as a series of events and student actions, using a sequence of static display images representing time slices during the scenarios. Judgments were elicited immediately following each scenario presentation, using a structured interview technique with a questionnaire. Likert-type ratings were used to assess the potential merit of the alternative view approach and specific features; questions with open-ended responses were used to identify issues, concerns and recommendations.

FINDINGS

The results of the research occurred in two parts: 1) instructional strategy and alternative view display designs, and 2) evaluation of the designs. Each are addressed separately below.

Display Design

The instructional strategy and aiding displays were designed around a hierarchical information structure, providing increasingly detailed information in successive layers under student control. The specific content was tailored to selected sonar training objectives under each of the two subject areas. The displays were configured around a "windows" format (i.e., assumed most readily familiar computer display format), and included several types of windows for information presentation, data entry and feedback. Elements of the training process strategy included:

- Alternative views of sonar console information, emphasizing graphics.
- Student control of aiding information access.
- Student had to actively request for increasing information details.
- Guidance information to help direct student actions.
- Information content was directed towards:
    - Identification of sonar/tactical parameters;
    - Parameter definitions and abbreviations;
    - Where-to-find data on sonar console;
    - Detailed guidance for procedures and processes;
    - Graphical representation of parameters;
    - Graphical representation of tactical aids;
    - Comparison of student-determined versus actual parameters;

Visualization of sonar phenomena.

Feedback information available at any point in the problem.

Student control design was for mouse-type of interface, or similar (e.g., joystick).

A representative frame of the Towed Array display is presented in Figure 1, showing the primary tactical display area and several nested information windows (both are noted in the figure).

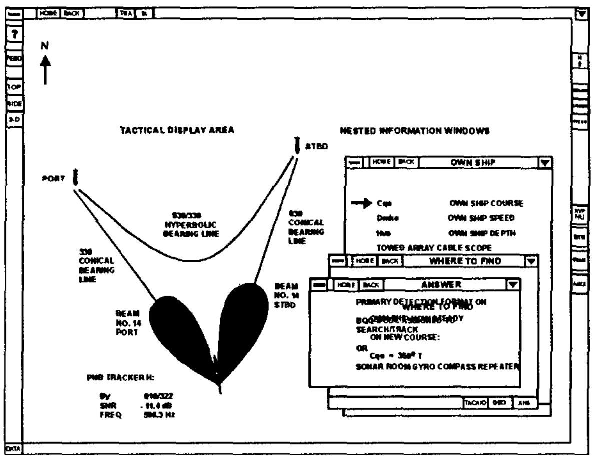


Figure 1. Towed array display example, including the Own Ship Information window, and Where-To-Find and Answer windows.

A second display example is presented in Figure 2, showing a Relative Motion/TMA instructional display with the Target information window open.

Both graphical and textual information is presented, as appropriate. When the student desires assistance as he progresses through the exercise, he would activate the normally blank aiding display located alongside the operational sonar console by moving the mouse. A bare-bones graphical display would be presented in the primary display area, organizing minimal problem-relevant information. The own ship-target information in the primary display area of Figure 2, without the Target information window overlay, is representative of the initial information detail.

The student would obtain additional information by selecting major elements on the display (e.g., clicking on the target icon). An information window overlay would open, such as the Target information window, presenting the next level of information. This next level of information would also be minimal, requiring student request for additional detail.

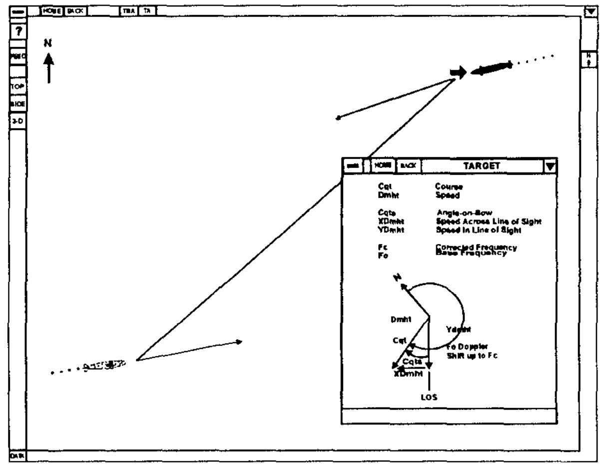


Figure 2. Relative motion/TMA display example, including the Own Ship Information window.

The student could descend to successively deeper, and more complete, levels of information by making successive requests. Additional information would be presented in the open windows, or additional windows would open, as appropriate to the problem and requests. The successive levels of information were designed in correspondence with the sonar problem solving process. For example: the hierarchical order of available information may be: identification of relevant sonar/tactical parameters, definition of acronyms, location of needed information (i.e., relevant parameters) on the sonar console displays, functional steps to calculate needed sonar/tactical quantities, representation of manual Tactical Aid computations, and resulting calculation answers.

The student could display additional information detail in any window when desired, by selecting displayed information and designating its destination (e.g., select target speed to be displayed as a vector in the primary display or in an information window). As the student manually calculates the sonar/tactical parameter quantities for the particular problem, as he normally would using grease pencil, he could enter them into the system, and enable comparative feedback information of calculated versus actual values.

The training process is envisioned to be tailored by the instructor, controlling availability of features and information. For example, access to answers may be allowed at any time early in the course, but constrained later in the course until all student calculations have been entered. Furthermore, the emphasis is to encourage student reliance on the operational sonar console for information, using the alternative view displays only when necessary, or to check on answers and progress. Student use of the assistance features would be automatically monitored, with summary information for the instructor to identify student difficulties.

Display Evaluation

The questionnaire collected data on overall judgments, as well as specific features and characteristics of the instructional approach and display design features.

Overall Potential Effectiveness. The responses to seven questions important to instructor/student judgment of potential overall effectiveness are presented in Figure 3, showing the mean rating for each question on a scale of 1 to 5 (1 = lowest, and 5 = highest rating) (Note, the expected mean was 3.0).

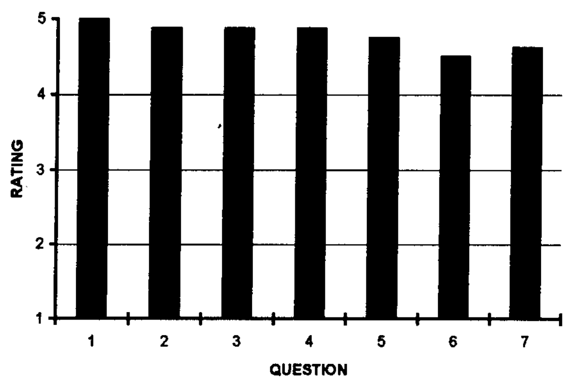


Figure 3. Sonar instructor and student overall judgments of the alternative view instructional approach.

The questions are:

1. Rate the overall effectiveness of the instructional aiding displays. (mean = 5.0, 95% Confidence Interval: 5.0 – 5.0).

2. Rate effectiveness of the displays for providing instructional assistance alongside an operational sonar console for individual training in the lab. (mean = 4.875, 95% Confidence Interval: 4.63 – 5.0).

3. Rate effectiveness of the displays for allowing the student to control access to instructional information -- when he wants it, and which information he wants. (mean = 4.875, 95% Confidence Interval: 4.63 – 5.0)

4. How does this type of display compare with the traditional trainer instructional methods, media and aids? (mean = 4.875, 95% Confidence Interval: 4.63 – 5.0)

5. How does this type of display compare with the traditional classroom instructional methods, media and aids? (mean = 4.75, 95% Confidence Interval: 4.43 – 5.0).

6. Rate the overall effectiveness of the relative motion/TMA instructional aid for assisting student learning and application of relative motion/TMA to sonar employment. (mean = 4.5, 95% Confidence Interval: 3.98 – 5.0).

7. Rate the overall effectiveness of the relative motion/TMA instructional aid for assisting student learning of bearing ambiguity resolution, and calculation of D/E angle. (mean = 4.625, 95% Confidence Interval: 4.11 – 5.0).

As evidenced by the data in Figure 3, the sonar instructors and students judged the potential value of the alternative view instructional aiding approach as very high, with the mean ratings all above 4.0. (Note, the lower confidence limits for 5 of the 6 means were also above 4.0, in comparison with the expected mean of 3.0). These high ratings express the need for instructional media that can assist in addressing the complex sonar concepts, as well as confirm the potential effectiveness of the alternative view approach.

Effectiveness of Specific Characteristics. The high approval ratings were due to characteristics of the media tailored to meet specific needs of sonar training, albeit in concert with guidance obtained from the research literature and the alternative view instructional approach. Examples of the instructor/student judgments of specific characteristics, also providing insight into display design features, are presented in Figure 4, addressing the following questions:

Questions 8 – 11. Rate the potential training effectiveness of this training aid in...

8. ...assisting the student to locate relevant information on the AN/BQQ-5 console. (mean = 4.63, 95% Confidence Interval: 4.11 - 5.0).

9. ...guiding the student to learn and perform sonar tactical calculations. (mean = 4.88, 95% Confidence Interval: 4.63 - 5.0).

10. ...guiding the student to learn and perform sonar tactical problems using the Tactical Aids. (mean = 4.88, 95% Confidence Interval: 4.63 - 5.0).

11. ...understanding sonar employment concepts and tasks. (mean = 4.63, 95% Confidence Interval: 4.11 - 5.0).

12. Rate effectiveness of top-down information access approach, providing increasing information detail with successive student requests. (mean = 5.0, 95% Confidence Interval: 5.0 - 5.0).

13. How important is the student entry of generated data? (mean = 5.0, 95% Confidence Interval: 5.0 - 5.0).

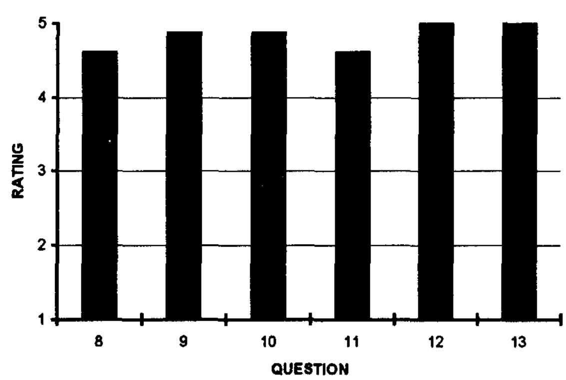


Figure 4. Judged effectiveness of example display characteristics.

The data presented in Figure 4 directly address several of the training process design elements identified earlier in this paper. The very positive instructor/student judgments emphasize the potential viability of these for sonar training, and approval for the implementation approach.

Certain features were not rated highly. For example, the mouse interface received widely varying judgments. Some instructors/students rated it highly, while others preferred another type of control, such as a trackball or joystick. Nevertheless, most features illustrated received high judgment ratings. This finding reflects the tailoring of instructional features designs to the identified needs of submarine sonar training.

Graphical Representation. The importance of providing visual representations of the underlying tactical representation along with procedural information was underscored by the large percentage of responses identifying visualization imagery (graphics) as the most important feature (i.e., 60% of responses). Improvements suggested by the instructors/students also emphasized visual imagery, with 36% of the suggestions pertaining to display design.

Other Findings. Other responses cited the instructional aiding approach as effectively addressing student needs by providing assistance information to students at the time they need it, and to the extent they desire. This approach would augment the instructor in a multi-student station laboratory, enabling applied individual student training using operational sonar consoles, and also providing instructional assistance to the level needed for each student.

The instructors/students were asked to judge the overall effectiveness of the strategy and displays for Basic, Advanced and LCPO sonar courses. High mean ratings were found for all courses (i.e., mean of 4.5 and above). They were highest for the Basic course, and decreasing in order for the Advanced course and the LCPO course. This suggests the training assistance is most needed when learning, or re-learning the sonar concepts and their applications. Once the concepts are well understood and the operators are proficient in their use, this type of assistance during training sessions is of less importance. This coincides with the training emphasis to use the operational displays as the primary information sources, and use the instructional displays only when necessary.

CONCLUSIONS AND RECOMMENDATIONS

The consistently high judgment ratings given to the overall training approach, and major and minor elements of the prototype displays, indicate the alternative view instructional approach has strong potential to enhance sonar employment training. The generic strategy and individual displays are appropriate in each of the sonar employment courses: Basic, Advanced and Leading Chief Petty Officer (LCPO). The implemented example features would, however, be most applicable to Basic sonar training.

The towed array and relative motion/TMA instructional displays appear to be reasonable prototypes for further development and dynamic evaluation during applied sonar training. Over three quarters of the features questioned received high judgment ratings, with the remainder receiving average to above average ratings.

Three features of the prototype displays stand out as having distinctive potential effectiveness for enhancing sonar employment training using the approach under development:

- Visualization imagery of the difficult-to-perceive sonar concepts;
- Emphasis on addressing and meeting student needs during the training process;
- An intuitive learning process tailored to the student, and sonar employment training.

The methodology of the alternative view approach (i.e., augmenting operational console displays, composed around major information windows, and keyed to the particular sonar problem) is expected to be an effective information access and presentation model for sonar employment training in the applied sonar laboratory context. The top-down information access structure, with nested widows for presenting the student-selected hierarchical levels of information, should be an effective strategy. The information presentation formats, combining graphical and textual information as appropriate to the exercise and student problem solving techniques in concert with the instructional strategy, should facilitate student use of the instructional assistance information.

The requirement for, and allowance of, direct student control over the training information access is an important component of the training approach, having received high approval. This results in the student as an active participant in the instructional process, and assists in focusing instructional assistance when and where needed.

The findings of this study demonstrate a need for incorporation of advanced instructional technologies into the applied training context, such as sonar employment training with operational consoles, to augment information provided on the operational equipment. This is emphasized for courses in which complex concepts and their application are being initially learned, and for courses in which they are being re-learned after students/operators had been away for some time.

The instructional strategy and displays, keying on familiar student cognitive representations and visualization imagery to assist in developing robust cognitive representations of the operational system, was found to have strong potential to improve the effectiveness of applied training dealing with difficult concepts. Implementation of a dynamic prototype of this instructional approach in applied sonar training is recommended to evaluate its training effectiveness.

REFERENCES

Cannon-Bowers, J.A., Salas, E. and Grossman, J.D. (1993). Improving Tactical Decision Making Under Stress: Research Directions and Applied Implications. Orlando, FL: US Naval Training Systems Center.

Hammell, T. J., and Ewalt, F. M. (1994). Sonar Trainer Instructional Display Design: Alternative View Approach. Report prepared for the Naval Air Warfare Center, Training Systems Division, Orlando, FL.

Klein, G.A. (1989). Recognition-Primed Decisions. In W.B. Rouse (Ed.), Advances in Man-Machine System Research (Vol. 5, pp. 47-92). Greenwich, CT: JAI Press.

Maxey, J.L., Scopatz, R.A., Madden, J.J., and Ahlers, R. (1993). Instructor Operator Station Training Aids: Preliminary Design Guidelines and Research Recommendations (Technical Report 92-013). Orlando, FL: Naval Training Systems Center.

Means, B., Salas, E., Crandall, B., & Jacobs, T. (1993). Training decision makers for the real world. In Klein, G., Orasanu, J., Calderwood, R., & Asambok, C. (Eds.), Decision Making in Action: Models and Methods (pp. 306-326). Norwood, NJ: Ablex.

Osgood, C. E. (1949). The similarity paradox in human learning: A resolution. Psychological Review, 56:132-43.

Pavio, A., Smythe, P.E., and Yuille, J.C. (1968). Imagery versus meaningfulness of nouns in paired-associate learning. Canadian Journal of Psychology, 22, 247-441.

Roscoe, S. N. (1980). Aviation Psychology. Ames, Iowa: The Iowa State University Press.

Weiten, W. (1992). Psychology, Themes and Variations. Pacific Grove, CA: Brooks/Cole Publishing Company.

THE RADAR SYSTEM CONTROLLER INTELLIGENT TRAINING AID

James E. McCarthy, Stephen Pacheco, H. George Banta, John L. Wayne, and David S. Coleman
Sonalysts, Inc.
Waterford, CT

ABSTRACT

The AN/SPY-1 is a phased array radar system that functions as part of the AEGIS combat system aboard modern U.S. Navy Cruisers and Destroyers. Enlisted personnel, known as radar system controllers (RSCs) operate and maintain the radar system. The RSC must optimize radar performance in a number of disparate environments. In order to enhance a new operator's ability to maintain this optimization, the AEGIS Training Center contracted for the development of a training aid. The resultant Radar System Controller Intelligent Training Aid (RSC ITA) is a PC-based training aid that makes use of a master/apprentice training paradigm. We describe it below.

ABOUT THE AUTHORS

Dr. James E. McCarthy has a B.S. in Mathematics Education from Winona State University, as well as an M.A. and a Ph.D. in Experimental Psychology from Miami University. He joined Sonalysts, Inc. in January 1993. Since that time he has designed and developed intelligent instructional technology for military and civilian uses.

Mr. Stephen Pacheco has a B.S. in Mechanical Engineering from the University of Rhode Island. He has over twenty years of experience as a lead software engineer. His work includes extensive experience with the design and development of simulation-based training systems.

Mr. H. George Banta has a B.E. in Civil Engineering from the Steven's Institute of Technology. He has a wealth of experience in the design and development of intelligent technology. Examples of his work include the development of intelligent systems to aid in resource allocation as well as tactical decision making.

Mr. John L. Wayne has a B.S. in Biology from Tulane University, and an M.B.A. in Management from Bryant College. He joined Sonalysts, Inc. in 1987 after 9 years as a surface warfare officer in the U.S. Navy. During his tenure at Sonalysts, he has assisted in the design, development, and utilization of a range of training systems.

Mr. David S. Coleman has a B.S. in International Security Affairs from the U.S. Naval Academy. Following seven years as a U.S. Navy surface warfare officer, he joined Sonalysts, Inc.. Since that time, he has had 13 years of experience managing a number of multimedia training system developmentefforts.

THE RADAR SYSTEM CONTROLLER INTELLIGENT TRAINING AID

James E. McCarthy, Stephen Pacheco, H. George Banta, John L. Wayne, and David S. Coleman
Sonalysts, Inc.
Waterford, CT

INTRODUCTION

Consider the various ways in which instruction could proceed. The most basic approach would be to disseminate the appropriate instruction in the form of a book. The book approach has many advantages. For example, books allow for self-paced instruction, support easy review of previously encountered content, and place the student in control of the content and order of instruction (see Laurillard, 1987 for a discussion of the benefits of student control). Unfortunately, books also have many disadvantages. A book presents information in only one style, and therefore is appropriate only for students with a compatible style of learning. Books cannot adapt to or interact with students. Books can only present static representations of the world and therefore are ill-suited to concepts that are fundamentally dynamic.

Another option would be to use classroom instruction. Classroom instruction is moderately adaptive and interactive. Students can exert some control over content, presentation method, and pacing through their questions and other interactions. Classroom instruction, however, has limits. The size of the class and (perhaps more importantly) the skill of the instructor determines, to a large extent, the amount of flexibility in areas such as instructional tactics and pacing. As class size increases, the likelihood of student or instructor initiated discussion decreases and it becomes more and more difficult for instructors to attend to individual students. Further, even if instructors were able to attend to individual students, they could only use those instructional tactics in which they had been trained and with which they felt comfortable. Student and teacher have little recourse if the instructor's repertoire of tactics does not include those which are most suitable for teaching a given concept to a given student. With large class sizes or with teachers with limited instructional skill or experience, classroom instruction suffers from all the negative attributes associated with book-based instruction without the advantages of self-pacing, content and order selection, or ready review.

Given the limitations of book- and classroom-based instruction, it may seem as if the ideal solution would be to assign to each student a tutor trained to teach the subject-matter in the style most appropriate for the learning style of the student. The tutor would be maximally adaptive to the student in that the pacing of instruction would reflect the student's ability to absorb the material and the method of instruction could be adapted to meet the moment-to-moment needs of the student. Even this seemingly ideal situation has limits, however. Besides the obvious cost disadvantage of the approach, human tutors, like books, are extremely limited in their ability to simulate complex dynamic events.

The preceeding discussion implies that the ideal training aid is a tutor that is pedagogically appropriate for each student, that is inexpensive to employ, and that can allow the student to interact with complex, reality-based, simulations. A simulation-based Intelligent Training Aid (ITA) is such a tool. Through careful construction, one can create a tutor that makes appropriate use of a wide range of instructional tactics, monitors the student, detects when he or she is experiencing difficulty, diagnoses the nature of that difficulty, and responds with remediation appropriate for both the difficulty encountered and the current state of the learner. This can be done in a complex, dynamic, and realistic problem-solving context that will allow the students to monitor the effect their appropriate and inappropriate actions have. The Radar System Controller Intelligent Training Aid (RSC ITA) described below is an example of such a system.

Reigeluth and Schwartz (1989) state that, "Computer-based simulations can provide efficient, effective, and highly motivational instruction that can readily serve the need for individualization. Simulations also enhance the transfer of learning by teaching complex tasks in an environment that approximates the real world setting in certain important ways." Others (Anderson et al., 1985; Burger, and DeSoi, 1992) echo the importance of hosting instruction within a problem solving context. Too often in contemporary education, theory is artificially separated from application. Simulation-based ITAs marry thought to deed and allow students to discover some truths on their own as they are coached to recognize others.

Before considering in detail the design of the RSC ITA, let's consider the context in which it will be used.

THE RADAR SYSTEM CONTROLLER: TASKS AND TRAINING

The AEGIS combat system is part of virtually all modern U.S. Navy Cruisers and an increasing proportion of U.S. Navy Destroyers (CG 47- and DDG 51-Class ships). The AN/SPY-1 radar system serves as the eyes of that system. The AN/SPY-1 is a phased-array radar system that automatically detects and tracks surface and air contacts. It then transfers the data to other portions of the combat system.

Enlisted personnel, known as radar system controllers (RSCs), are responsible for operating and maintaining the radar system. RSCs must manage the radar system to minimize the temporal or spatial blindness facing the combat system. Temporal blindness refers to the length of time it takes the radar system to search the battle-space. Spatial blindness refers to the range at which radar detects and tracks a contact of a given size. Increasing spatial blindness reduces temporal blindness. Conversely, decreasing spatial blindness generally results in an increase in temporal blindness. The RSC must maintain a balance between spatial and temporal blindness that is appropriate for the existing tactical environment.

RSC training takes place during a 24-week radar system operations and maintenance course. Maintenance training occupies the first 23 weeks of that course. Operations training occurs during the last week of the course. During the laboratory portion of operations training, each student operates a functioning radar system for approximately 3 hours.

Feedback and lessons-learned from Operation Desert Shield/Storm and other fleet operations revealed that novice RSCs who completed the training course were, in general, ill-equipped to deal with the complex management task facing them in the fleet. Consequently, the AEGIS Training Center sought to improve the level of operational training provided to RSCs by contracting for the development of a training aid that would allow RSC-trainees to practice their operational skills in an operationally realistic training environment. The Radar System Controller Intelligent Training Aid (RSC ITA) is the result of that development effort.

INTELLIGENT TRAINING AIDS: OVERVIEW

As a rule, all intelligent training aids share four common components: a learning environment/ student-device interface, a domain expert, a student model, and an instructional expert. Let's consider each of the components in turn.

Learning Environment/Student-Device Interface

The learning environment defines the context in which learning takes place. The learning environment defines the tasks facing the student as well as the tools the student can use. The student-device interface is the medium of communication between the student and the ITA.

ITA developers are increasingly taking the perspective that students learn from doing. The perspective stems from an instructional philosophy that holds that learning is a process of construction not absorption, and that new ideas must be tied to each other and to old ideas (Burton, 1988, Burger & DeSoi, 1992). Learning environments now more than ever require students to apply and even discover knowledge.

A critical dimension of the learning environment and student-device interface is the degree of similarity or fidelity between the learning environment and the real world. Burton (1988) identified four kinds of fidelity: physical, display, mechanistic, and conceptual. It has been noted that it is important to match the level of fidelity to the training task at hand. For example, if students are learning sensorimotor tasks, then it is likely that the ITA needs high levels of physical fidelity. On the other hand, if we are teaching reasoning skills, conceptual fidelity is probably more important than physical fidelity.

Together, the choice of an appropriate learning environment and student-device interface scheme are critical for the success of an ITA. An effective scheme can, in fact, enhance learning before the intelligent components of the software ever come into play.

Domain Expert

The domain expert is a software module that represents the knowledge or performance of someone expert in the domain of instruction. For example, if we were to build a medical diagnosis ITA, the domain expert would represent the knowledge of an expert diagnostician. In the present case, the domain expert reflects the performance of an expert RSC.

There are three general classes of domain experts. They are: black box, glass box, and process models of expert decision making (Anderson, 1988). Black box models of expert decision making are usually extremely efficient algorithmic processors that produce the correct input/output behavior in the instructional domain. Black box domain experts produce the correct solution and therefore they can judge the correctness of the student's actions. However, because they use processes that are unlike those used by human experts, they cannot produce instructionally useful explanations of their behavior. Moreover, although the black box expert can judge the correctness of a student's action, the judgment cannot be extended to include diagnosis of the student's difficulty or misconception.

Although black box domain experts have acknowledged weaknesses, they can support lower-cost development of intelligent training aids (e.g., Gugerty & Hicks, 1993; Gugerty, Hicks, and Walsh, 1993). As these examples point out, black box expert systems can be re-purposed from other tasks to support a form of intelligent instruction. Black box models eliminate the need for extensive knowledge engineering efforts thus reducing the time and cost of development. They do so, however, with a price of reduced capability.

The second class of domain experts are glass box models. Glass box models solve problems in the domain by using reasoning heuristics that are similar to those employed by human experts. Glass box domain experts can judge the correctness of a student's actions, they can diagnose a student's difficulty, and they can provide explanations of the expert's decision. Constructing glass box domain experts require more extensive knowledge engineering efforts. As a result, they are generally more capable, but more expensive.

The third class of domain experts are process models (or cognitive models). Process models attempt to encode and employ knowledge in human-like ways. The benefit to this approach is that the knowledge is in a form that the system can most easily and completely communicate to the student. The cost is that process models are relatively more difficult and expensive to construct. Process models must commonly take the form of production systems (e.g., Anderson, et al., 1985) or semantic networks (e.g., Carbonell, 1970). Process models can judge the correctness of student actions, explain the expert's solution to a problem, identify related issues, generate real-time remediation, and support mixed-initiative dialogues with the student.

In considering the class of domain expert to include in an intelligent training aid, the developer must weigh the cost of development against the cost of sacrificing power and elegance. As always, the correct solution is a function of the constraints of the problem.

Student Model

The student model is the intelligent training aid's conception of who the student is. The student model is the repository of data that allows the ITA to adapt to the particular student using the system. Generally, each student has his or her own student model and each model is updated as the students interact with the ITA.

There are four classes of student models: Performance models, overlay models, error models, and simulation models (Ohlsson, 1986). The fidelity of the associated domain expert constrains the selection of a student model.

Performance models focus on *how much* the student knows, not *what* the student knows (Ohlsson, 1986). Performance models are the only type of student model that is available when we use a black box model domain expert. Performance models allow us to assess the global level of understanding of the student. With them, we can adjust problem difficulty or pacing. They do not, however, contain enough data to permit us to tailor instruction to the particular difficulties facing the student.

Overlay models represent the student's knowledge as a sub-set of an expert's. Overlay models present an expert's knowledge as a collection of concepts. They then record the student's mastery, or lack thereof, of each concept. Overlay models are most consistent with glass box domain experts. The overlay model allows us to determine specific areas of strength and weakness for each student. This knowledge allows the ITA to use the students' strengths to overcome their weaknesses.

Error models represent the student as possessing some number of common misconceptions. The misconceptions are often called "bugs". Therefore, error models are often called "bug catalogues". Error models are most consistent with glass box and process models of the domain expert. Because they point out specific shortcomings in student performance the ITA can adapt instruction to combat those weaknesses.

Simulation models represent the student as following a more or less appropriate problem solving script. In some sense, simulation models represent a combination of overlay and error models. The student's correct and incorrect actions are combined to form a simulation of that student's performance. Simulation models are most consistent with process models. As with overlay and error models, simulation models permit instruction tailored to specific areas of strength and weakness.

Note that each of the classes discussed above focuses on student knowledge, not on student cognitive style. A complete student model should include information on both. Two factors have hampered efforts in this area. The first is that there are very few tests of specific cognitive style dimensions. The second is that educational psychology has made very few prescriptions regarding how to teach a given concept to a student with a particular cognitive style (Ohlsson, 1986). In the absence of such prescriptions, cognitive style data is of little use to the ITA and therefore student models have not incorporated it. For ITAs to approach their potential, developers and educational psychologists must address these issues.

Instructional Expert

Just as the domain expert is a software module that represents the knowledge of an expert in the instructional domain, the instructional expert is a software module that represents the knowledge of one skilled in instructional practice.

Of the all ITA components, the instructional expert is the least formalized. The situation stems from the observation made in the preceding section: educational psychology has not yet been able to suggest how to teach a particular student a given concept in a defined context. In the absence of such guidelines, the design of instructional experts has tended to be ad hoc and quite variable across systems.

Instructional experts use data from the student model together with data from the most recent domain expert assessment of student actions to make instructional decisions. If the student model is performance based, the instructional expert must decide whether to change the pacing of instruction and/or the difficulty of the problems facing the student. When a richer student model is available, the situation becomes more complex.

When the ITA has an overlay, error, or simulation student model, there are three decisions an instructional expert must make. First, the expert must decide whether or not to intervene in an instructional situation. Many times skilled instructors observe a student making a mistake but remain silent so the student can discover the error for himself or herself. Similarly, if the student is performing correctly, the expert must decide whether more is gained through positive reinforcement than is lost by interrupting the student.
If the instructional expert decides that the present situation warrants intervention, it must then decide which issue or concept to discuss. Often a single student action will reveal multiple instances of student knowledge and misconception. The instructional expert must choose one to address.

Finally, the instructional expert must choose the form of the instructional intervention. All ITAs have a number of ways to present remediation on a given topic. The instructional expert must choose the form of remediation that is most beneficial for the particular student at that time.

Let's now consider the RSC ITA in terms of each of these components.

RADAR SYSTEM CONTROLLER INTELLIGENT TRAINING AID

Overview

The RSC ITA is a master/apprentice (Burger and DeSoi, 1992) training system that uses the pedagogical principle of fading. The RSC ITA attempts to foster a synthetic master and apprentice relationship similar to the one that exists in a shipboard environment. When a novice RSCs arrive on board, they must satisfy personnel qualification standards before they are allowed to stand watch alone. The standards are satisfied under the watchful eye of an experienced RSC who monitors the novices' actions and provides instruction as needed. The RSC ITA functions in a similar manner. Figure 1 depicts the general functioning of the RSC ITA.

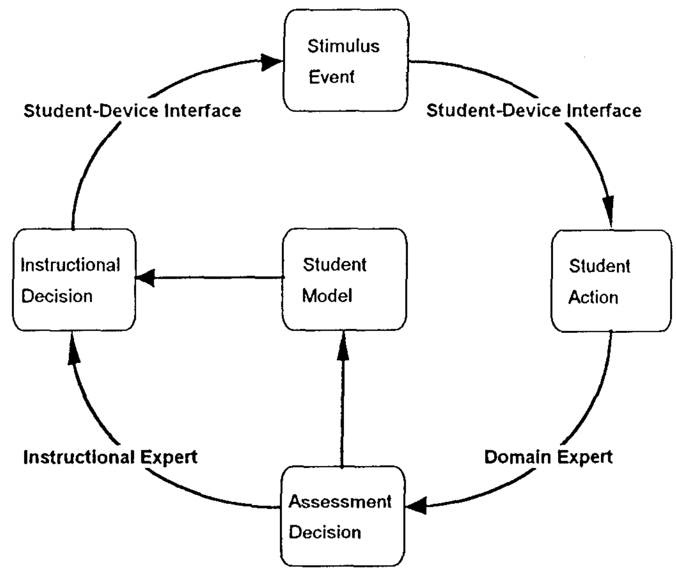


Figure 1: A Cycle of Interaction

For the most part, the RSC ITA is a simulation-based training aid. Therefore, most interaction begins with some sort of simulation-based stimulus event. The event must then be conveyed to the student. That is the job of the student-device interface. Next, the student takes some action in response to the stimulus event. The master, or domain expert, that is figuratively looking over the student's shoulder evaluates the student's action. The evaluation leads to an assessment decision which, in turn, is used to update the student model. The instructional expert then uses data from the revised student model and the assessment decision to reach an instructional decision. The student-device interface then conveys the decision to the student and the cycle repeats itself.

Now, let's consider each component of the RSC ITA in some detail.

Learning Environment/Student-Device Interface

The learning environment and student-device interface is an extremely important component of the ITA. We made every effort to create an operationally realistic learning environment. We paid special attention to attaining high levels of cognitive fidelity in the design of the student-device interface and the learning environment.

The learning environment is a functional replication of the RSC watchstation. It includes a model of the world, a model of the radar system, and a simplified replication of the operator's console. The world and radar models interact to produce symbology on the console's PPI and a dynamic amplitude spectrum on the console's A-Scope. Clutter, jamming, and hostile and friendly surface and air tracks are all recreated. As the student changes the radar configuration, the appearance of these entities is affected. For example, increasing sensitivity will tend to increase the detection range of a given target.

The student-device interface, depicted in Figure 2, is a Windows™-based reproduction of the OJ-451 console display and controls used by the RSC. Controls are activated by standard point-and-click operations.

In response to simulation events, the student could take a range of actions. For example, the student could hook tracks (i.e., select them for closer inspection), build sectors and sub-sectors, impose radar doctrine, observe the effects of doctrine, and communicate with other combat information center (CIC) team members. Watching all student activity is a

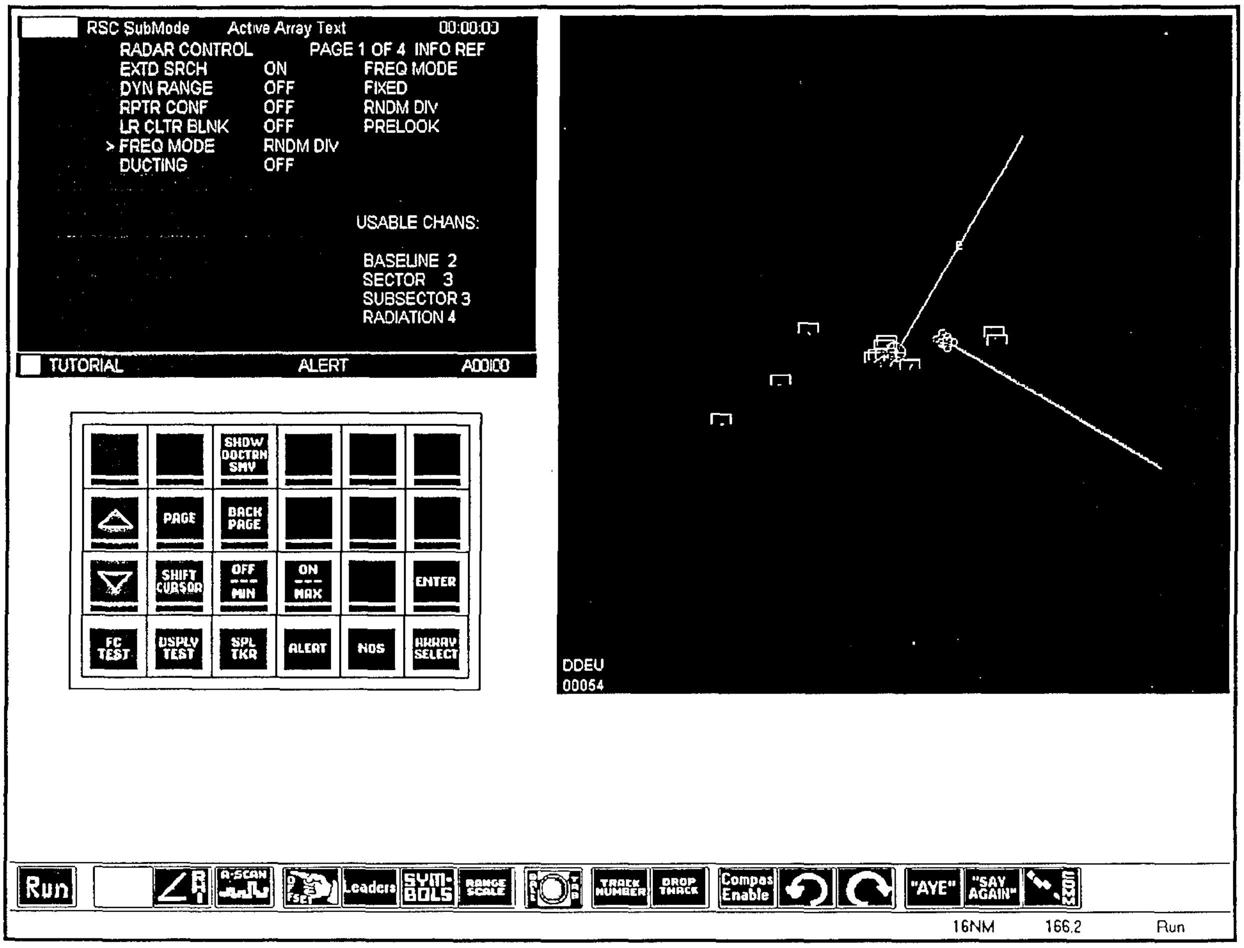


Figure 2: RSC ITA Student Device Interface

domain expert module in the guise of a master RSC.

Domain Expert

The domain expert in the RSC ITA is based on a process model. One segment of the domain expert observes the console and reaches conclusions about the proper course of action. For example, the domain expert might observe that the time required to search the battlespace was approaching unacceptable levels. At the same time, it might note an isolated region of false (clutter) tracks. The domain expert might then decide that it was necessary to impose doctrine in the clutter region. The domain expert would also know in which order to try various doctrine impositions (i.e., it would know which "fixes" to try first and which to try last), as well as how long to observe each imposition to see if it was successful. Finally, the domain expert would be able to recognize when it had acheived a satisfactory radar configuration.

The domain expert's decisions set up a number of expectations. As the student performs, the domain expert gathers the actions and compares them to the existing expectations. Three classes of actions might result:

1. The student's actions might meet the domain expert's expectations.
2. The student's actions might represent a mutation of the expected action (i.e., a bug). For example, the domain expert might expect the student to build a particular sector, and the student might build the sector, but enter the bearing limits incorrectly.
3. The student might take an action that is totally unexpected, that is, it does not conform to any of the expectations nor the anticipated mutations of those expectations.

The comparison of the expectations and actions leads to an assessment decision and the revising of the student model.

Student Model

The student model assesses mastery of three types of knowledge: declarative (i.e., knowledge of facts), procedural (i.e., knowledge of actions), and contextual (i.e., knowledge of timing). An endorsement tree (Murray, 1991) serves as the underlying formalism.

The endorsement tree formalism recognizes that often there are multiple sources of information about a student's mastery of a given concept, and that the sources differ in their reliability. For example, a student's claim of knowledge and his or her demonstration of knowledge both are informative, but we are more likely to believe the demonstration than the unsubstantiated claim. Further, the endorsement tree formalism recognizes that knowledge of a student's mastery of one concept can tell us things about the student's mastery of other concepts. For example, if a student has demonstrated mastery of fraction multiplication, then it is likely that the student has mastered integer multiplication. Finally, the endorsement tree formalism rejects numeric methods (i.e., confidence levels or weighted averages) in favor of symbolic methods (i.e., counting within ordinal equivalence classes).

To construct the student model, we first decomposed the knowledge domain into a hierarchy of learning objectives. For example, a learning objective such as, "The student can build a low-power sector", has as a component, "The student can build a sector", which, in turn, has as a component, "The student can enter sector bearing limits". The decomposition led to the construction of a learning objective tree. Next the sources of evidence for and against mastery were defined. Examples include default beliefs, inherited or propagated beliefs, answers to single questions, data trends, and activities which reflect knowledge. The evidence sources are used to define classes of evidence that are ordered by their assumed reliability.

The student model places each evidence datum that the domain expert passes to it in the appropriate evidence class. When a decision based on the student's mastery of a concept is required, the student model applies the following procedures. Beginning with the most reliable evidence category, the student model pairs positive and negative endorsements. If there are more positive endorsements, the student model assumes that the student has mastered the objective. If there are more negative endorsements, the student model assumes that the student has not mastered the objective. If there are an equal number of positive and negative endorsements, the student model moves on to the next most reliable evidence class. If at the end of the process, all positive evidence is balanced by an equal amount of negative evidence, the student model labels mastery uncertain. If no endorsements exist, the student model labels mastery unknown.

If the instructional expert needs a finer grain of data on which to base its decision, the endorsement model can provide a strength of belief value as well. The strength of a belief is based on the evidence class on which it is based, as well as the number of positive or negative arguments that remain after pairing. In addition, the instructional expert can, conceivably, make use of the full pattern of data in all evidence categories.

An endorsement-based student model can support an instructional expert as it matures and becomes more complex.

Instructional Expert

In light of any clear prescriptions on its construction, we decided to design the instructional expert with an eye towards simplicity and ease of expansion.

The first decision the instructional expert must reach is whether or not to intervene. When the instructional expert makes the decision, it must balance its need to be unobtrusive with the need to forestall poor operating procedures. The RSC ITA instructional expert only intervenes when the student model posts a new endorsement. The student model only posts endorsements after the student has completed a meaningful block of sections (e.g., after the student has pressed the "Enter" key to enter all the settings pertaining to a radar sector). If the endorsement is positive and the student has a high level of mastery on that objective, the instructional expert intervenes (with positive feedback) only if it has not made a recent positive feedback comment to the student. If the endorsement is positive and the student has demonstrated a lack of mastery, the instructional expert always intervenes. If the endorsement is negative, the instructional expert always intervenes.

Next, if the student model records more than one new endorsement, the instructional expert must decide which learning objective to address. If all the endorsements are positive then the instructional expert picks the most general learning objective for positive reinforcement. If multiple positive endorsements exist at the same level, the instructional expert chooses the one discussed least recently. If there are negative endorsements, the instructional expert chooses to discuss those. If there are multiple negative endorsements, the instructional expert chooses the most specific learning objective. If there are multiple negative endorsements at the same level, the instructional expert chooses the one discussed most recently.

Finally, the instructional expert must select the form of the intervention. The ITA uses the instructional philosophy of "fading". The instructional expert instantiates the philosophy by providing more information to students whose level of mastery is low and less information to those students whose level of mastery is high. For example, assume that two students, one with a low level of mastery and the other with a high level, both performed the same correct action. On one hand, the instructional expert might say something like "Good Job!" to the high level of mastery student. On the other hand, the instructional expert would probably tell the low level of mastery student what he did, that it was correct, and why it was correct. The low level of mastery student needs this level of information to tune his performance, but the high level of mastery student would probably find it intrusive.

Conclusion

Intelligent training technology provides a powerful training alternative in both the military and civilian sectors. Intelligent training aids such as the RSC ITA allow students to apply skills in realistic ways as they acquire them. Application-based instruction is entirely consistent with constructionist learning theory and, in all likelihood, results in greater retention and transfer of training.

The master/apprentice training paradigm used within the RSC ITA is extremely flexible and can be applied to a number of training problems. Simply put, the master/apprentice paradigm is appropriate whenever the training task could best be accomplished through direct work-related tutoring. There are countless examples in both the military and civilian sectors of such settings. The examples include learning to diagnose system faults in power plant operations, learning to operate military sensor and weapons systems, and learning engineering principles in an academic setting.

Although a number of research issues still exist in this field, none is more striking than the need for educational models that support the development of instructional expert modules. Developers of intelligent trainers must work together with educational psychologists to insure that intelligent training systems reach their full potential.

Bibliography

Anderson, J.R. (1988). The expert module. In M.C. Polson, & J.J. Richardson (Eds), Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Anderson, J.R., Boyle, C.F., and Reiser, B.J. (1985). Intelligent tutoring systems. Science, 228, 4698, 456-462.

Burger, M.L. and DeSoi, J.F. (1992). The cognitive apprenticeship analogue: A strategy for using ITA technologyfor the delivery of instruction and as a research tool for the study of teaching and learning. Int. J. Man-Machine Studies, 36, 775-795.

Burton, R.R. (1988). The environment module of intelligent tutoring systems. In M.C. Polson, & J.J. Richardson (Eds), Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Carbonell, J.R. (1982). An investigation of computer coaching for informal learning activities. IEEE Transactions of Man-Machine Systems, 11, 190-202.

Clancey, W.J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman & J.S. Brown (Eds.), Intelligent Tutoring Systems. New York: Academic Press.

Gugerty, L., & Hick, K. (1993). Non-diagnostic intelligent tutoring systems. Proceedings of the 15th Interservice/Industry Training Systems and Education Conference, November, Orlando, FL.

Gugerty, L. Hicks, K., & Walsh, W. (1993). From an intelligent job aid to an intelligent-computer-aided training system: Training applications of the Integrated Maintenance Information System (IMIS). Proceedings of the 15th Interservice/Industry Training Systems and Education Conference, November, Orlando, FL.

Laurillard, D. (1987). Computers and the emancipation of students: Giving control to the learner. Instructional Science, 16, 1, 3-18.

Murray, W.R. (1991). An endorsement-based approach to student modelling for planner-controlled intelligent tutoring systems. Technical Paper, AL-TP-1991-0030. Air Force Systems Command, Brooks Air Force Base, Texas.

Ohlsson, S. (1986). Some principle of intelligent tutoring. Special Issue: Artificial intelligence and education. Instructional Science, 14, 3-4, 293-326.

Reigeluth, C.M., and Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. Journal of Computer-Based Instruction, 16, 1, 1-10.

Multiship Simulation as a Tool for Measuring and Training Situation Awareness

Wayne L. Waag, PhD
USAF, Armstrong Laboratory
Mesa, Arizona

ABSTRACT

In 1991 the USAF Chief of Staff posed a series of questions regarding situation awareness (SA) in fighter operations including the following. Can SA be measured? Can SA be trained? This paper presents the findings of a research investigation that explored the use of networked multiship simulation as a tool for measuring and training SA. The Division's MULTIRAD simulation facility was used which permitted two F15s to fly against a suite of manned and unmanned adversaries in a realistic combat environment. Controller support was provided using a long-haul network linked to an AWACS simulation located at Brooks AFB, TX. A week-long evaluation syllabus was designed consisting of 9 sorties with 4 engagements per sortie. A building block approach was taken so that scenarios increased in difficulty over the week. Sixty-three mission ready F15 pilots participated in the study. Critical incident/event data and performance ratings of SA were gathered using two trained observers. Additionally, mission outcome, network communications, video recordings, and eye movement data were gathered. As expected, SA was found to be related to previous experience with Fighter Weapons School graduates, as a group, performing the best. Performance was found to improve for identical engagements flown early and late in the syllabus. Positive opinions were expressed by study participants regarding the potential value of multiship simulation for training SA skills. Areas of greatest payoff appear to be the training of flight resource management and decision-making skills. It was concluded that multiship simulation can be an effective tool for both measuring and training SA.

ABOUT THE AUTHOR

WAYNE L. WAAG is currently the Senior Scientist at the Aircrew Training Research Division of the Armstrong Laboratory located at Williams Gateway Airport (formerly Williams AFB), Mesa, Arizona. Upon completion of his doctoral work in 1971 at Texas Tech University in experimental psychology, he received a Postdoctoral Research Associateship from the National Academy of Science/National Research Council with tenure at the Naval Aerospace Medical Research Laboratory at Pensacola, Florida. In 1973, he accepted a position with the Flying Training Division of the Air Force Human Resources Laboratory at Williams. He has remained at Williams until present with the exception of an exchange scientist tour from 1979 through 1981 with the United Kingdom's Royal Air Force. Dr Waag is currently leader of the Armstrong Laboratory's Situation Awareness Integration Team (SAINT) which cuts across multiple directorates and divisions. He is also team leader of the Divisions current research program into combat situation awareness training. His personal research interests lie in the area of measurement and modeling of aircrew behavior with emphasis on tactical decision making under conditions of stress.

Multiship Simulation as a Tool for Measuring and Training Situation Awareness

Wayne L. Waag, PhD
USAF, Armstrong Laboratory
Mesa, Arizona

INTRODUCTION

This paper presents some preliminary findings of an attempt to use multiship, air combat simulation as a tool for both measuring and training situation awareness (SA). It is part of a larger research investigation conducted by the Armstrong Laboratory of SA within the F-15 fighter community.

The impetus for the present investigation came directly from the US Air Force Chief of Staff. In 1991, he posed a series of questions concerning situation awareness (SA) within the operational F-15 fighter world. First of all, What is SA? Can it be objectively measured? Is SA learned or does it represent a basic ability or characteristic that some pilots have and others do not? In response to the question, "what is it?" a working group at the Air Staff produced the following operator's definition of SA: "a pilot's continuous perception of self and aircraft in relation to the dynamic environment of flight, threats, and mission, and the ability to forecast, then execute tasks based on that perception (Carroll, 1992)." While other definitions of SA within the literature focus primarily on processes underlying the assessment and resulting knowledge of the situation (Endsley, 1988; Fracker, 1988), our working definition also included forecasting, decision making, and task execution. From an operational Air Force perspective, SA is more than simply knowledge and understanding of the environment.

The Armstrong Laboratory subsequently initiated a research investigation that had three goals: first, to develop and validate tools for reliably measuring SA; second, to identify basic cognitive and psychomotor abilities that are associated with pilots judged to have good SA; and third, to determine if SA can be learned, and if so, to identify areas where cost-effective training tools might be developed and employed.

To develop measurement tools (the first goal of the study), it was first necessary to identify and describe critical behavioral indicators of the fighter pilot's ability to maintain good SA and successfully complete his mission. To this end, Houck, Whittaker, and Kendall (1993) conducted a cognitive task analysis of a typical F-15 air combat mission. The resulting analysis identified the significant types of decisions required of the flight members, the information required for making these decisions, and the observable activities the flight members performed to acquire this information. The results were further analyzed by an experienced fighter pilot to identify behavioral indicators considered most essential to SA. This subject matter expert (SME) emphasized that these behavioral indicators must be observable in the context of day-to-day squadron training activities and subject to evaluation by fighter pilots both in terms of their own performance and that of others. As a result of this analysis, 24 behavioral indicators organized in seven categories were identified and are shown in Table 1.

Based principally upon these behavioral indicators, a number of SA Rating Scales (SARS) were developed to measure SA in operational units. They were administered to 238 mission-ready F-15 pilots from 11 operational squadrons. From the SARS, a composite measure of SA was derived and found to be highly related to previous flight experience and current flight qualification (Waag & Houck, 1994). These measures were used for two purposes. First, they served as a criterion measure against which to validate a battery of basic ability tests considered relevant to SA, thereby addressing the question of basic human abilities (the second goal of the study). The Situation Awareness Assessment Battery (SAAB), consisting of 24 computer-based tests of basic cognitive and psychomotor abilities (Carretta, Perry, & Ree, 1994), was also administered to the same sample of pilots at their home units.

1. TACTICAL GAME PLAN
   Developing plan
   Executing plan
   Adjusting plan on-the-fly
2. SYSTEM OPERATION
   Radar
   Tactical electronic warfare system
   Overall weapons system proficiency
3. COMMUNICATION
   Quality (brevity, accuracy, timeliness)
   Ability to effectively use information
4. INFORMATION INTERPRETATION
   Interpreting vertical situation display
   Interpreting threat warning system
   Ability to use controller information
   Integrating overall information
   Radar sorting
   Analyzing engagement geometry
   Threat prioritization
5. TACTICAL EMPLOYMENT-BVR
   Targeting decisions
   Fire-point selection
6. TACTICAL EMPLOYMENT-VISUAL
   Maintain track of bogeys/friendlies
   Threat evaluation
   Weapons employment
7. TACTICAL EMPLOYMENT-GENERAL
   Assessing offensiveness/defensiveness
   Lookout
   Defensive reaction
   Mutual Support

Table 1. Behavioral Indicators and Categories of Performance

Second, these measures served as a means of selecting a sample of pilots who participated in a simulation phase of the effort, in which performance was observed under realistic combat conditions. During this phase, simulated air combat mission scenarios were developed for assessing SA and a variety of performance measures gathered in an attempt to determine whether SA could be measured in simulation environment. Moreover, an attempt was made to examine the potential of this type of simulation for training critical SA skills. This paper presents some preliminary findings of the data gathered from a simulated air combat environment.

METHOD

Subjects

A total of 40 mission-ready (MR) F-15 pilots, who were flight lead qualified served as subjects. An additional 23 MR F-15 pilots served as wingmen throughout the data collection which began in Mar 93 and was completed in Jan 94.

Simulation System

The Armstrong Laboratory multiship simulation facility (MULTIRAD) located at Williams Air Force Base (WAFB), Arizona (now Williams Gateway Airport, Mesa, AZ) was used. The major components of the simulation system are shown in Figure 1. These components represent independent subsystems operating as part of a secure distributed simulation network. This local area network was connected to the air weapons controller simulator (AESOP) at Brooks Air Force Base (BAFB), TX by a dedicated T-1 telephone line. Additional details concerning the basic simulation architecture and components are available in Gehl, Rogers, Miller, and Rakolta (1993) and Platt and Crane (1993).

The manned flight simulators consisted of two

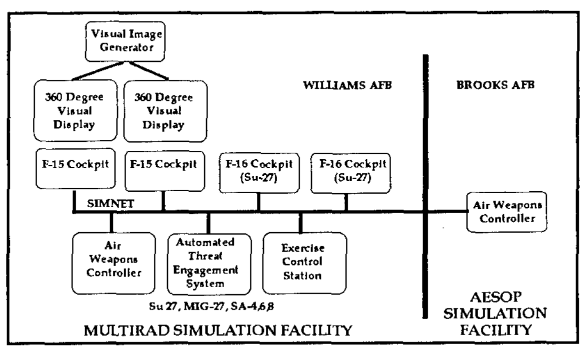


Figure 1. Multiship Simulation Facility

F-15C simulators and two F-16 simulators. The F-15C simulators had high fidelity aerodynamic, engine, avionics, radio, sensor, and weapons simulations. Each F-15C simulator was equipped with an out-the-window visual display system covering approximately 360 deg horizontal by 200 deg vertical. The external visual scene was created using computer-generated imagery. The manned F-16 simulators had less fidelity and played the role of enemy aircraft in conjunction with computer-controlled adversaries. The visual and electronic signatures of these F-16 simulators were modified so that they appeared as the appropriate threat aircraft. Each F-16 simulator was equipped with a single channel of out-the-window visual imagery covering approximately 45 deg horizontal by 45 deg vertical.

A manned air weapons controller (AWC) provided the F-15C pilots with appropriate threat information and warnings. Depending upon the availability of qualified AWCs and equipment status, the AWC was either located at WAFB or BAFB. In either case, the AWC had a realistic simulation of the appropriate AWC console and communicated with the F-15C pilots by radio.

The exercise control system (ECS) consisted of a central console with the hardware and software necessary to create, start, observe, record, and stop the simulated air combat sorties. The SMEs who served as test directors and observers viewed monitors that provided a real-time view of each sortie. These monitors provided: 1) a plan view display of all the participants in each engagement along with status information; 2) the instrument panel of each F-15C cockpit which included the radar, radar warning receiver, and armament displays; and 3) the forward channel of out-the-window video for each F-15C cockpit. The plan view display, instrument panel displays, and radio communication were also recorded to video tape for mission debrief and further data analysis. In addition, the ECS included a data logger that recorded all the network communication protocols between simulators.

Ground threats, as well as additional threat and friendly aircraft, were provided by a computer-based automated threat engagement system (Rogers, 1992). The ground threat portion of the automated threat engagement system (ATES) provided command and control functions (e.g., early warning radars and target assignment) and simulation of directed and autonomous surface-to-air missile batteries and anti-aircraft artillery with their radars. The aircraft portion of the ATES provided computer controlled air interceptors as well as formations of air-to-ground bombers. In addition, the ATES provided four computer controlled F-16s which were escorted by the manned F-15Cs during offensive counter air sorties.

Scenario Design

The primary approach taken toward the measurement of SA was through scenario manipulation and observation of subsequent performance as recommended by Tenney, Adams, Pew, Huggins, and Rogers (1992). Other approaches such as the use of explicit probes (Endsley, 1988) were considered and finally rejected due to their lack of face validity for the study participants. Since we were using mission-ready F-15 crews, it seemed essential that we provide a simulation experience as realistic as possible. A week-long SA "evaluation" exercise was constructed that consisted of 9 sorties with 4 engagements per sortie. Sorties were arranged in a building block manner. Over the week, engagements increased in complexity in terms of numbers of adversaries, enemy tactics, lethality of ground threats, AWC support, etc.

A typical engagement scenario is presented in Figure 2. This depicts a defensive counter air (DCA) mission in which the objective of the two F-15s is to defend the home airfield. In this case, the attackers consist of two bombers accompanied by two fighters. The engagement begins at 80 nautical miles (nm) separation in which the fighters are flying at 20,000 ft. and the bombers at 10,000 ft. They are laterally separated by 10 nm which makes them fairly easy to acquire on radar by the two F-15s. At 35 nm, the fighters begin a corkscrew type of maneuver in which they rapidly descend to 3500 ft. At this time, they will drop off of the F-15s' radar screen. Upon completion of the maneuver, the fighters will trail the bombers as well as being at a much lower altitude. While the F-15s can easily continue tracking the bombers, it requires the crew to "predict" the actions of the fighters so that they may be quickly re-acquired on radar. At 15nm, the bombers do a hard right turn and descend to 2500 ft. At this time, the bombers will momentarily drop off the radar screen. Since the range is very close (10-12 nm), it requires the crew to accurately "predict" the actions of the bombers and correctly use their radar so that they may be quickly re-acquired. The problem is further complicated in that the bombers and fighters will now "merge" in roughly the same airspace. If the fighters are ignored, then they can launch against the F-15s. If the F-15s "lock" their radar on the fighters, which will usually be the

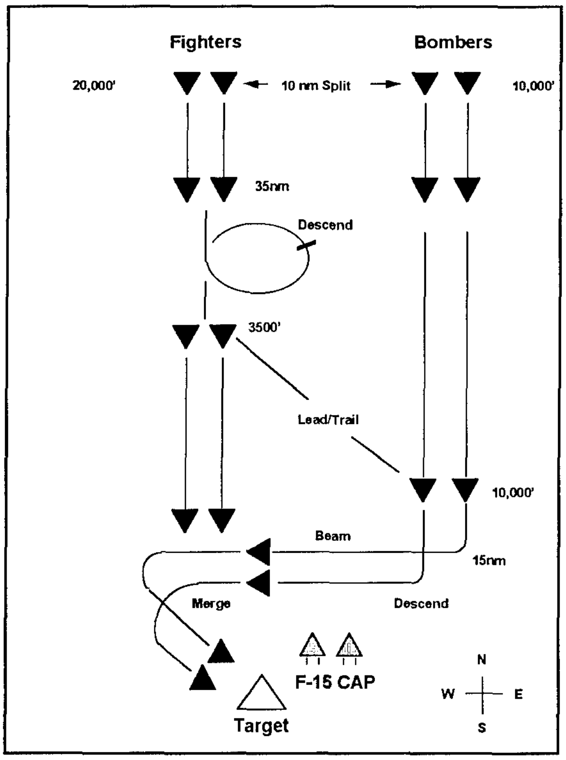


Figure 2. Typical Engagement Scenario

case at this point, then the bombers can continue toward the airfield "untargeted." Once the fighters are engaged, it is very difficult to re-acquire the bombers since they are low and will be flying away from the F-15s. If the F-15s fail to kill the fighters, the problem will only be compounded.

This example not only shows the approach taken toward the design of the mission scenarios, but also serves to illustrate our contention that SA is more than knowledge of the current situation. In operational environments, situation assessment and decision making are viewed as tightly coupled and are often difficult to separate. For the fighter pilot to be successful, he must not only be able to "build the big picture," but he must also translate his assessment into an employment decision. Often, the inability to make these critical employment decisions may lead to mission failure, despite a correct assessment of the situation. In the sample scenario, the key to success is to target and destroy the bombers prior to 15 nm and then target the fighters. If the ranges become so close that all four threats must be dealt with simultaneously then the mission is likely to fail. It is through the careful design of such mission scenarios that the failure to incorrectly assess the situation or make incorrect employment decisions can be successfully inferred based upon the observation of pilot performance in the unfolding of the mission scenario.

Data Sources

Given the tremendous cost of gathering data on MR F-15 pilots, the approach was to gather as much as possible from a variety of sources. In our view, the most important data sources were the judgments and observations of two retired fighter pilots who possessed an in-depth understanding of the air combat domain. The same two SMEs were used throughout the year-long data collection effort. For each mission, the following procedure was followed. One of the SMEs would attend the mission briefing session conducted by the crew. During the conduct of each mission both SMEs observed mission performance. One of the SMEs also served as the mission director who was responsible for starting and stopping each engagement, communicating with the console operator, etc. During each engagement, each SME independently completed an observational checksheet to record pertinent events, notes, and outcomes. Upon completion of the four engagements comprising a single mission, one of the SMEs accompanied the crew to the debriefing room. The flight lead was responsible for conduct of the debriefing, although the SME was permitted to ask questions in an attempt to clarify the crew's understanding of the situation and purpose of their actions. Upon completion of the debrief, the two SMEs discussed each engagement, and completed a consensus performance rating scale consisting of the 24 behavioral indicators of SA related to F-15 mission performance. The SMEs also produced a written critical events analysis for each mission which attempted to identify those events that, in their opinion, affected the outcome of the mission and were indicative of the crew's SA.

A variety of other data were also gathered. These included mission events and outcomes such as weapons firings, kills, etc. Using the data logger in the ECS, the digital data passed over the network was recorded, whereby each engagement could be reconstructed. The videos recorded and used for debriefing were also archived. Additionally, eye movement data were recorded for the four engagements flown on the last mission. And finally, all participants were also asked to "critique" the simulation and also give opinions regarding its potential for training.

RESULTS

The results from two data sources are presented in this paper, the performance ratings from the two SMEs, and the critiques regarding the potential value of the simulation for training. These data are used to address the two issues central to this paper, namely, and use of simulation as a tool for both measuring and training SA.

Simulation As an SA Measurement Tool

One of the original goals of the overall research program was to develop techniques for measuring SA. In essence, two approaches were taken; first, the development of SA rating scales that could be administered within the operational units; and second, the development of techniques based upon observed performance within a controlled simulation environment. To briefly summarize the first approach (Waag and Houck, 1994), three SA Rating Scales (SARS) were developed to measure pilot performance in an operational fighter environment. These instruments rated SA from three perspectives: supervisors, peers, and self-report. SARS data were gathered from 238 mission-ready USAF F-15C pilots from 11 operational squadrons. Reliabilities of the SARS were quite high as measured by their internal consistency (.95 to .99) and inter-rater agreement (.88 to .97). Correlations between the supervisory and peer SARS were strongly positive (.89 to .92), while correlations with the self-report SARS were positive, but smaller (.45 to .57). A composite SA score was developed from the supervisory and peer SARS using a principal components analysis. The resulting score was found to be highly related to previous flight experience and current flight qualification. In fact, this score was used as the basis for selection of pilots to participate in the simulation phase of the effort that is described here.

One question of interest is the relationship between the SA scores based upon peer and supervisor ratings in the squadron and the SA scores derived from the simulation environment. The hypothesis was that there would be a moderately positive correlation between these two sets of scores. Simply stated, pilots judged to perform very well in the units should also perform well within a controlled simulation environment and vice-versa. Mean performance ratings given by the SMEs across the four engagements were computed for each mission. These mission ratings were then regressed against the single score obtained from the units. A scatterplot of these data are presented in Figure 3. The resulting correlation was found to be .56 ($p<.01$).

These results support the hypothesis of a

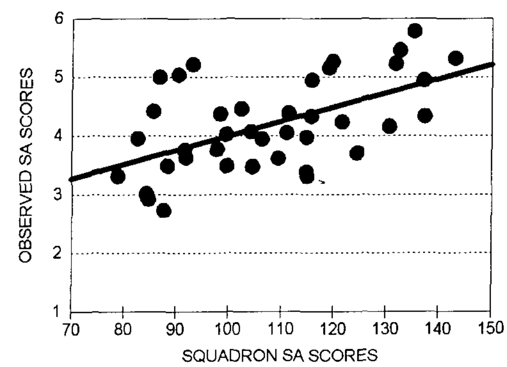


Figure 3. Scatterplot of SA Scores in Squadron Versus SA Scores in Simulator

relationship between SA as measured in both environments. They also indicate that although the relationship is positive, it is not perfect.

Simulation As an SA Training Tool

The other issue concerned the potential of multiship simulation as a tool for training SA skills. Given the definition of SA that was adopted at the outset of the study, this question translates into the issue of whether training in this type of simulated combat environment transfers to the real airborne environment. While transfer is an easy concept to understand, it is extremely difficult to measure given the enormous costs and complexities of carrying out such evaluations.

Bell and Waag (1994) have proposed a five-stage sequential evaluation model for conducting training effectiveness evaluations. In order, these include: (1) utility evaluation; (2) in-simulator performance improvements; (3) transfer to alternative simulation environment; (4) transfer to a flight environment; and (5) extrapolation to a combat environment. The authors made use of a multiship combat simulation similar to that used in this study as a vehicle for discussion of the requirements of each of these stages. The data gathered from the study presented bear only upon the first two--user opinion and performance improvement.

Two types of user opinion data were gathered--ratings of the training benefit for various pilot experience levels and an open-ended questionnaire. The results of the ratings of potential training benefits are provided in Figure 4. These data clearly indicate that positive opinions were expressed by the study participants on the value of this type of simulation for training. The

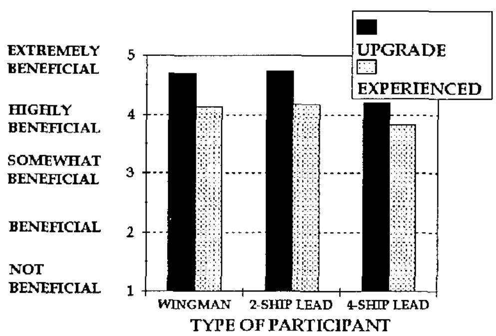


Figure 4. Rated Benefit of Training for Various Levels of Experience

potential training was considered beneficial for all levels of qualification. However, as expected, greater benefit would be expected for pilots upgrading into a given qualification level.

Opinions expressed in the open-ended questionnaire were also quite positive. Although qualitative, they provide additional insight into the potential focus of training using multiship simulation and how it might be employed. In particular, mention was made of using such training as a means of enhancing both situation assessment and decision-making skills. It was also frequently noted that there was tremendous value in learning flight leadership and resource management skills. In terms of the location of such simulation, the overwhelming consensus was that they would be of most value within the operational units. This was not too surprising since each unit now has the operational version of the cockpits used in the present investigation. However, they are stand-alone and non-visual, and as such their training capability is fairly limited. In contrast, the networking of such devices within a realistic combat environment increases the potential greatly. The bottom line from the utility data is that the participants considered multiship simulation as a tool with high training potential.

While positive user opinion is a necessary prerequisite for effective training, in itself, it is insufficient validation (Bell & Waag, 1994). At the next stage of the evaluation model, it is necessary to demonstrate improved performance within the simulation environment as a function of practice. In other words, it is necessary to show that learning has occurred. It should be pointed out that it was never the intent, at the outset of the study, to demonstrate performance improvements. It must be emphasized that the sole purpose was to develop a set of simulation scenarios that could be used to assess SA within a combat environment. As such, normal training interventions were not permitted. For example, during the debrief, pilots were permitted to only view their own in-cockpit displays and not the planned view display. Moreover, the two SMEs were not permitted to provide any type of feedback to the pilots regarding their performance.

However, data from the ninth mission did permit some comparison since identical scenarios had been flown earlier in the week. The ninth mission was designated the "eye track" mission in which eye movement data was recorded. For these scenarios, an eye tracker computed point of gaze and was displayed against the background scene as determined from a scene camera mounted on the pilot's helmet. The resulting video signal replaced the second cockpit display within the ECS. This permitted the crews to debrief the final mission using three integrated displays, the planned view of the fight, their own cockpit display, and the eye-tracked display which portrayed point of gaze against the background scene. Although not central to this paper, it should be mentioned that very positive opinions were expressed by the pilots regarding the potential of eye movement recordings as a feedback tool for training. It was viewed as potentially useful for the earlier stages of training and, in particular, for the diagnosis of problems of students encountering difficulty. It could potentially provide a solution to the continuing problem of training for single-seat aircraft in which instructors complain that diagnosis is difficult when one cannot see where the student is looking.

Two scenarios, a 2 V 2 defensive counter air (DCA) mission and a 2 v 4 offensive counter air (OCA) mission, were flown during the middle of the week and then again on the last mission. A comparison of performance is presented in Figure 5. In both cases, performance on the last mission was improved. However, only the 2 V 2 DCA mission was found to be statistically significant.

It should be recalled that the scenarios were designed to increase in difficulty over the week.

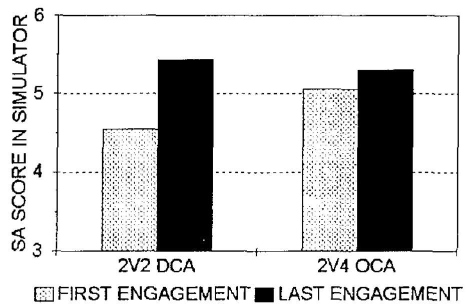


Figure 5. Effects of Practice on Observer SA Ratings

Consequently, if one simply plots the Observer SA Scores across missions, there is generally a downward trend. To obtain an estimate of what the curve might look like assuming "equal difficulty" of all scenarios, a magnitude estimation procedure was undertaken to scale the difficulty of the scenarios. Raters included the two SMEs and another in-house F-15 pilot who had occasionally served as wingman in the course of the study. Only missions 2 through 8 were included since mission 1 was a "familiarization" sortie and mission 9 was the eye track sortie. These difficulty weightings were then applied to the mean observed SA scores for each mission. The results are presented in Figure 6.

It is clear that when the scores are weighted for scenario difficulty, the resulting curve suggests that performance improved over the week. Again, it should be cautioned that the procedures followed were not the most appropriate for a conduct of a rigorous test of learning within the

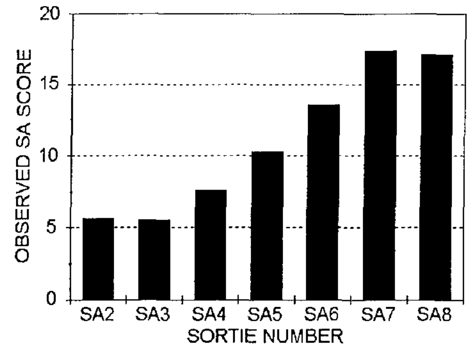


Figure 6. SA Scores Weighted for Scenario Difficulty Across Missions

simulation environment. However, when such data are coupled with the very strong pilot opinions that they had received valuable training, it seems reasonably safe to conclude that learning had occurred over the week.

DISCUSSION

This study attempted to answer two questions. Can multiship simulation be used as a tool for both measuring and training SA? Each of these is discussed. The reader should keep in mind the operational definition of SA that was adopted at the outset of the investigation since it does markedly differ from others that have been used.

First, can multiship simulation be used as an assessment tool? In my view, the answer is clearly "yes." The data presented in this paper show a positive relationship between SA as measured within the operational units and SA as measured in a controlled simulation environment. Although the relationship is positive, it is not perfect. The data from the units were found to relate very strongly to previous flight experience and current flight qualification. In general, the same relationships were observed in the simulation data, although their magnitudes were reduced. Those pilots with more flight hours and a higher flight qualification, in this case an instructor pilot rather than a 2-ship flight lead, generally performed better. As a group, the best performers were those pilots who were weapons officers, indicating that they were Fighter Weapons School graduates. Taken as a whole, these data suggest that those pilots with more experience tend to perform better within a controlled simulation environment.

However, there occurred noticeable exceptions to this general trend. For example, consider the three pilots in Figure 3 who had low squadron scores but performed extremely well in the simulation environment. These individuals were fairly inexperienced two-ship leads and for that reason obtained low squadron scores. However, these individuals adapted extremely well to the demands of the scenarios that were used in the study. In other words, they learned very quickly and adapted to the demands of the combat environment. In fact, their performance was superior to other pilots who were certainly more experienced. It should be emphasized that the scenarios flown on the last four missions were of a complexity that is rarely experienced within operational training environments due to resource constraints. Although speculative, such data suggest that simulation may be a useful tool in assessing not only current performance, but also predicting who is likely to excel in new environments for which they have not received training.

Second, can multiship simulation be used as a training tool? In my view, the answer is, again clearly "yes." From a user's perspective, the data are very clear regarding the potential value of such simulation for training. The 63 MR F-15 pilots overwhelmingly considered such training to be of value. Although such anecdotal evidence is often considered suspect from a scientific perspective, it is nevertheless an absolute prerequisite for effective training. Unless there is user acceptance, the resulting training will be of marginal value regardless of the device's inherent potential.

In addition to the opinion data, there is evidence that performance did improve within the simulation environment; in other words, learning did occur. Again, it should be pointed out that the amount of improvement was probably "minimized" due to the evaluative orientation of the investigation. When identical scenarios were flown early and late during the week, the performance on the second repetition was better. Additionally, when scenario difficulty is assumed constant, the resulting weighted scores show improvements. These data combined with the fact that the study participants expressed opinions to the effect that their proficiency had improved leave little doubt that learning had occurred.

Although the data clearly indicate (1) that the end user expresses very positive opinions toward the value of multiship simulation and (2) that learning occurs, there still remains the issue of transfer to the real world which represents the "acid test." Clearly, the data gathered in this study do not bear upon that issue. For the "believer," evidence to date is strong enough to warrant the conclusion that training will be effective. In fact, given the previous transfer of training research that has already been conducted (Waag, 1981; Bell & Waag, 1994) there is little reason to suspect that such training within a multiship simulation environment would not have a positive effect upon subsequent performance in the air. Yet, for the "skeptic," no definitive evidence has been presented.

CONCLUSIONS

Based upon the findings of the present study, it is concluded that multiship simulation can be successfully used as a tool for both measuring and training SA. Future efforts should focus upon the development of appropriate training strategies and interventions which will maximize its training potential.

REFERENCES

Bell, H.H. & Waag, W.L. (1994). Evaluating the Effectiveness of Flight Simulators for Training Combat Skills: A Review. Submitted to International Journal of Aviation Psychology.

Carretta, T.R., Perry, D.C., & Ree, M.J. (1994). Prediction of situational awareness in F-15 pilots. Submitted to International Journal of Aviation Psychology.

Carroll, L.A. (1992). Desperately seeking SA. TAC Attack (TAC SP 127-1) 32: 5-6.

Endsley, M. R. (1988). Design and evaluation for situation awareness enhancement. In Proceedings of the Human Factors Society 32nd Annual Meeting (pp. 97-101). Santa Monica, CA: Human Factors Society.

Fracker, M.L. (1988). A theory of situation assessment: implications for measuring situation awareness. In Proceedings of the Human Factors Society 32nd Annual Meeting (pp. 102-106). Santa Monica, CA: Human

Factors Society.

Gehl, T.L., Rogers, R.L., Miller, M.A., & Rakolta, J. (1993). Interdependence of training utility and network performance using the Armstrong Laboratory Multiship Research and Development System. In, Proceedings of 15th Industry/Interservice Training Systems Conference, Orlando, FL: National Security Industrial Association.

Houck, M.R., Whittaker, L.A., & Kendall, R.R. (1993). An information processing classification of beyond-visual-range intercepts. (AL/HR-TR-1993-0061). Brooks AFB TX: Armstrong Laboratory.

Platt, P. & Crane, P. (1993). Development, test, and evaluation of a multiship simulation system for air combat training. In, Proceedings of 15th Industry/Interservice Training Systems Conference, Orlando, FL: National Security Industrial Association.

Rogers, B. (1992). Tactical air threat system for a distributed simulation network. In, Proceedings of 14th Industry/Interservice Training Systems Conference, San Antonio, TX: National Security Industrial Association.

Tenney, Y.J., Adams, J.J., Pew, R.W., Huggins, A.W.F, & Rogers, W.H. (1992). A principled approach to the measurement of situation awareness in commercial aviation (NASA Contractor Report 4451). Langley VA: National Aeronautics and Space Administration.

Waag, W.L. (1981). Training effectiveness of visual and motion simulation. (AFHRL-TR-79-72). Brooks AFB, TX: Air Force Human Resources Laboratory.

Waag, W.L. & Houck, M.R. (1994). Tools for assessing situational awareness in an operational fighter environment. Aviation Space and Environmental Medicine. 65 (5, Suppl.):A13-A19.

SYSTEMS ENGINEERING AND ARCHITECTURE: LESSONS FROM THE F-22 TRAINER PROGRAM

Tony DalSasso
F-22 System Program Office
Wright-Patterson AFB, Ohio

George R. Rovny
Lockheed Fort Worth Company
Fort Worth, Texas

ABSTRACT

The successful implementation of a training simulation system requires that engineering constraints be communicated from the Systems Engineer to the designers in an unambiguous manner. This paper proposes that an architectural framework can be developed, providing the Systems Engineer with a tool to aid in this communication.

The paper documents the F-22 Pilot Training System team's observation that the term "architecture" has no universally-accepted definition. It chronicles the process used to resolve this problem, eliminating the confusion concerning both the terminology and the process of developing an architecture. It describes a hierarchy of definitions, allowing consensus to be reached among a group with widely varying experience levels, without creating a "least common denominator" definition. It explains the term by means of analogy - what "architecture" means to a builder, and how this maps into the trainer engineering context.

Emphasis is given to how an architecture needs to address hardware and software as a system. A preferred process for creating the architecture and managing the subsequent development of the product, using architecture as a systems engineering tool, is discussed. The paper describes the "litmus test" developed to determine whether an approach constitutes an architecture and describes the attributes of an architecture that allow its relative quality to be measured. It observes that most of what is touted as architecture doesn't pass the "litmus test," and why it does not.

Believing that the F-22 program is a microcosm of a trend throughout industry, the paper suggests that lessons learned by the F-22 can be effectively applied elsewhere. It discusses why this subject is so vital to a simulation development effort, and concludes with some thoughts on how a properly developed architecture can provide significant advantages to a system integrator.

BIOGRAPHY

Tony DalSasso is employed by the U.S. Air Force, Air Force Materiel Command, Aeronautical Systems Center, F-22 System Program Office, Wright-Patterson AFB, Ohio. He is presently serving as Lead Engineer for the F-22 Pilot Training System. He has previously served as Lead Engineer on the Standard Simulator Data Base Program (Project 2851), and participated in various training simulator acquisition programs, including the B-2 Aircrew Training Devices and Special Operations Forces Aircrew Training System, specializing in the areas of visual simulation and terrain data bases. He holds a BS in Computer Engineering from Lehigh University.

George Rovny is a Lead Engineer at the Lockheed Fort Worth Company, formerly the Fort Worth Division of General Dynamics. Mr. Rovny has been with Lockheed since 1984, holding various engineering positions in avionics integration and test, design of electronic support systems, simulation, and training systems. Mr. Rovny has been involved with the F-22 Training System since its Demonstration/Validation phase and currently has lead technical responsibility for the Lockheed contribution to the Pilot Training System. Mr. Rovny holds a Master's Degree in Electrical Engineering from the Pennsylvania State University.

SYSTEMS ENGINEERING AND ARCHITECTURE: LESSONS FROM THE F-22 TRAINER PROGRAM

Tony DalSasso
F-22 System Program Office
Wright-Patterson AFB, Ohio

George R. Rovny
Lockheed Fort Worth Company
Fort Worth, Texas

INTRODUCTION

As part of the Air Force's implementation of Integrated Weapon System Management, the F-22 Advanced Tactical Fighter program includes the development and deployment of a complete training system for aircrew and maintenance personnel. The development of the training system is proceeding concurrently with that of the aircraft, allowing for the early exchange of design information among the system of Integrated Product Teams (IPTs) developing the F-22. Early involvement allows training system requirements to be considered during the development of the air vehicle, supporting the incorporation of features in developmental software which would facilitate the future reuse of this code in the trainers; it also gives the Training System IPT detailed knowledge of the air vehicle design process.

Through the progress of the program, it has become apparent that the integration of reuse items from the aircraft will be difficult, if not impossible, in the absence of a defined hardware/software framework for the trainer. This is particularly true of the Pilot Training System (PTS), which requires a significant amount of functionality beyond that found in the engineering labs. There are enough differences among the anticipated operational configurations of the aircraft, avionics engineering laboratories, and training system to cause concern that the initial concept of using the laboratory simulation architecture in the PTS will provide a substandard solution at excessive lifecycle cost.

With this philosophy, the F-22 program team set out to lay the groundwork for the establishment of a training system architecture. The first hurdle to be overcome was the lack of a common understanding of the terminology: "exactly what do you mean when you say 'architecture'?" This drove the team to develop a working definition, which turned out to be an iterative learning process, involving team members of varying disciplines. The definition of architecture, and its relationship to the systems engineering process, will form the basis of future work on the F-22 Training System.

ARCHITECTURE AND SYSTEMS ENGINEERING

What is Architecture?

Prior to relating the experiences of the F-22 program in developing a working definition of architecture, it is first necessary to understand the concept. Early on, it was discovered that considerable ambiguity surrounded the term. It was found that the definition of the word "architecture" seems to be quite controversial, probably not because it is an especially difficult concept to grasp, but more likely because nearly everyone *thinks* they understand the term already. Unfortunately, there is rarely consensus among individual interpretations; this leads to a breakdown in communication and creates controversy. The F-22 program solved this problem by establishing a common interpretation, as discussed herein.

General Definition

Webster's Ninth New Collegiate Dictionary includes five definitions for the word *Architecture*. The one which is most applicable in the systems engineering context is probably definition 2b, "a unifying or coherent form or structure." Before examining the engineering implications of this term, let us first understand the more familiar use of this definition, associated with the external appearance of buildings.

Construction Analogy

Everyone seems to understand the term "architecture" when applied in this context; so it is a good starting point for our discussion. Architecture is used to describe certain properties associated with a physical structure. Indeed, it is such a fundamental property that practitioners of building design are referred to as "architects"[1].

The architecture of a building might be described as how its individual components are arranged to give a characteristic appearance to the structure as a whole. In this sense, architecture is used to establish such unifying properties as proportion, materials, and ornamentation. Using just these key attributes, even the layman can differentiate among Greek, Colonial, Spanish, Gothic, Modern, and the various other architectural styles.

It is important to understand that architecture has little to do

[1] Calling this person an "architect" can actually lead to a misunderstanding of the term, insofar as very few "architects" actually develop an architecture. Under this definition, the building architect is a designer, applying architectural principles rather than developing them.

with functionality. Although certain architectural styles are often associated with buildings used in a particular way - a Gothic church, for example - it is not an inherent property of the architecture that makes this so; rather, the application of a given architecture to some construction problem is at the discretion of the building designer. The designer does not select an architecture based upon specific functional requirements, but rather on the basis of some unifying aesthetic theme which he desires the resulting structure to possess. Any building of sufficient size can be used as a church; nothing about its functionality requires that it be of Gothic design, and in fact the majority of churches are not built in that style. Conversely, structures of many different functions can share a common style. One could build a Gothic home, office, or factory, if desired.

Architecture vs. Design. "Architecture" should not be confused with "design" - they occur at different phases in the development of a structure. Architecture is effectively a constraint imposed upon the design process. It is a set of "ground rules" which guide the development of a design. Thus, the design of a particular building does not constitute the *development* of an architecture; it is, in fact, the *application* of an architecture. Architecture builds a repository of information for later use by individual designers.

Philosophically, one could think of the development of an architecture as a creative, "right-brain" activity, whereas the development of a design is an analytic, "left-brain" task. Architecture establishes the set of components and the rules for connecting them; design applies these materials and tools to create a functional entity. It is conceptually easy to see how the former is a creative process, while the latter is more or less mechanical. In general, it's easier to follow rules than it is to make them.

There are many benefits of having an architecture as a precursor to a design. For one, relative to methodology, creativity is expensive. Risks are always greater when one starts with a clean slate, rather than adhering to established techniques. Architecture allows the lessons of the past to be applied to the problem being investigated. Architecture allows a design problem to be solved but once.

Relationship to Engineering. But how does this relate to "architecture" as defined in the context of engineering? Consider the concept that architecture creates a set of components necessary to solve an entire class of problems, and that design applies these parts to create the solution to a specific problem. In construction, the facility designer takes architectural elements and arranges them to configure a functional structure. Engineering is equivalent, in that the designer starts with these fundamental "building blocks," and applies them to the problem space. In hardware, this equates to an engineer designing a board around a standard chipset, and interfacing to a standard bus. The preponderance of low-cost PC hardware attests to the validity of this process.

When one examines the design process, one can see that the design of a building is not significantly different than the engineering process used to build a trainer, or any other engineered product. The Systems Engineer (SE) - or system architect[2] - creates processes and templates; the design engineer, or implementer, uses them to create the product. The implementation is divorced from the creation of the rules which guide it. Just as the building designer is relieved of the responsibility of creating the system of proportions, materials, and ornamentation which define the style of his structure, the design engineer is free to concentrate on the functional aspects of his product, with the knowledge that the architecture, properly applied, will yield a unified result.

Building architecture does not unduly constrain the designer in functional terms - designing a church in the Gothic style does not have any impact on its utility as a church, for example. Similarly, the application of an engineering architecture does not limit the designer's control over the functional capabilities of the device. But it allows the designer to achieve a predictably unified result, while avoiding the difficulty associated with starting the design from scratch. In short, architecture makes the designer's job easier, while it assures the quality of the product.

Cost Management

If architecture were to be distilled down to its very essence, its sole *raison d'être* would be cost management. Architecture gives the SE the ability to divide the problem into a set of manageable sub-problems, the implementation of which can be delegated to individual design engineers. It gives him visibility into the development process, and control over it. When it becomes time to integrate the components into a unified product, he can be confident that they will integrate in the least amount of time. During the development phase of a system, all of these activities take time, which equates to dollars. Minimizing the time spent on any of these functions translates into savings to the program. Further, any "unknown" can impart a risk to the program, and the potential that a greater amount of time will be spent than originally anticipated. The architecture allows risk to be controlled, which results in the control of time and, in turn, control of the cost of the system.

The cost control aspects of architecture are not limited to the development phase, but can extend well into the lifecycle of the product. By enhancing the understandability of the design, the maintenance of the system can be performed by the fewest number of personnel; this will hold support costs down. With a consistent architecture, changes to the system will be easier to make, minimizing the amount of time spent in their analysis and implementation. Again, this will result in a cost saving over the lifecycle.

2 In this case, the title of "architect" is applied in a manner consistent with our chosen definition.

Through its establishment of standards for developers, architecture can be a much more effective means of controlling the cost than less formal methods, such as handing off sets of requirements to developers who then proceed to design their subsystems independently. The latter case gives the SE little or no control over the quality of the individual subsystems, which will likely succumb to the creative whims of their developers. Architecture controls cost through the selective management of creativity. Designers at the lower levels are empowered to apply creativity only within their specific problem domains, and within the constraints imposed by the architecture and their specific standards.

What is Systems Engineering?

Systems Engineering is defined by the Defense Systems Management College [DSMC 90] as "the management function which controls the total system development effort for the purpose of achieving an <u>optimum balance of all system elements</u>. It is a process which transforms an operational need into a description of system parameters and integrates those parameters to <u>optimize the overall system effectiveness</u>."

Systems Engineering is a process by which a solution framework is created from some problem statement, the solution is communicated to designers, and the realization of the solution by designers is governed. The SE must understand the nature of both the Engineering/Manufacturing Development (EMD) phase and lifecycle constraints, the classes of platform technology available (including hardware, compilers, and toolsets), and must have a sense of the standards in use by the designers. The latter is important as the SE must strike a balance in the solution between the degree of specificity (level of constraint) and the degree of latitude afforded the designers. The balance will depend on the degree of confidence the SE has in the accepted level of standards in the design community.

Expanding on lifecycle constraints, in the world of Pilot Training devices, it is typical that the lifecycle will be characterized as being long and subject to numerous requirements changes, coming from changing user needs as well as changes to the air vehicle. Changes will always have to be accommodated very rapidly. The SE must devise solutions that accommodate frequent and rapid modification.

<u>Process.</u> The SE can ensure robust solutions if he bases the structure of the system on the structure of the problem. That is, he will define and bound a "problem space", and extract from it pertinent features, creating an abstract representation of the problem definition that will yield a set of patterns. The conceptual framework for the solution should be based on the structure of these patterns, which can be made to incorporate the desired attributes of the system. The design decisions must be conveyed to the developers; the SE uses the architecture as the primary vehicle to communicate the design rules. The SE monitors and guides the development process, through integration and test, enforcing the architecture and arbitrating conflicts as they arise. He will search for emerging problem areas and attempt to address them long before significant resources are expended.

There is a troubling tendency on a large program to embrace the prevailing standards and methodologies of the day, and the massive organizations that are created to enforce them, as the *de facto* Systems Engineering process. While programs are often required to utilize one methodology or another (as F-22 is), it is useful to recognize that these methodologies are often attempts to regain control over projects that have scaled beyond the limited utility of methods that worked for small systems, and that the methodologies represent more rigid, disciplined application of techniques that worked for small systems. The message here is simply that blind trust of methodologies to produce complex systems will result in disappointing failure. Methodologies don't replace Systems Engineering.

<u>The Training System Development Challenge.</u> When applied to the problems of Training Systems, the Systems Engineering process must deal with constant uncertainty and persistent information voids. During EMD, requirements are likely to remain unstable until late in development, as the mission and the air vehicle are in a constant state of flux. And the validation process at the end of EMD is not as straightforward for training devices as it is for many other products (especially in the case of F-22, where the contractor is to validate the creation of an effective "Training System", not a device that does x, y, and z). Training System development has been described using the "Q-Tip model," i.e. the longest part of the development process, in the middle, is fairly firm and easy to grasp, but there is fuzz at both ends.

These factors create a fascinating environment for the Training Device Systems Engineer. He faces a nebulous problem on the left, tempted to view the Instructional System Development (ISD) process as esoteric mysticism, while on the right he is constantly warned by the pragmatic designers that their primary metric used to divert the wrath of management is one called "requirements volatility".

At this point it is useful to recall the postulate observed earlier, that "architecture has little to do with functionality". Neither is its basic form affected much by fidelity requirements. So the seemingly impossible situation just described provides no excuse for the SE to despair. The SE must forge strong ties with the instructional analysts, to extract from them requirements that are meaningful in an engineering context, while alerting them to capabilities of the technology as well as the air vehicle that they may not have been privy to. He must be prepared to provide rapid analysis when asked by instructional analysts what the cost of various capabilities will be in various device types. Solid Systems Engineering can magnify the value of ISD.

The SE must forge a solution that easily accommodates frequent change, minimizing both the cost and schedule impact of any change. He should provide to the designers a framework whose very structure embodies attributes amenable to change, isolating designers from the volatility inherent in Training Devices.

A Training Device SE thus bridges the gap between the instructional analysts and the developers, ensuring that function and fidelity are manifest within a robust, coherent, highly maintainable system by guiding the development process according to the architectural solution.

Role of Architecture

The architecture of a system is by far the most important tool available to a SE. With it he can predict and manage cost, comprehend system behavior, govern the creativity of developers, and communicate with the multiple entities who have a stake in the project.

Cost Management. Architectures have become increasingly important to SEs as projects have become more costly to develop and more costly to maintain. The SE will utilize the architecture to predict and manage the cost of a large project that requires a large staff. This implies that the architecture will express the structure of a system at a sufficiently fine level of granularity that the cost of individual pieces can be reasonably estimated. An argument exists that the smaller the partitions are, the less subjective the cost projection will be.

Comprehend System Behavior. It has been stated that the purpose of architecture is to give the SE intellectual control over a system. The SE will use the architecture to predict system behavior in a variety of scenarios, and will utilize this understanding to render decisions for developers as ambiguities are discovered. As system characteristics emerge during development that call into question the validity of assumptions made to develop the architecture, the SE will modify it as needed to capture that new understanding.

It has proven advantageous to understandability for the partitioning of a training simulation to exhibit a close mapping to the structure of the article being simulated for training. The SE will search for patterns in the air vehicle to serve as the basis for the partitioning of the training devices.

The architecture provides the basis for thought experiments, whereby the system can be stressed by numerous scenarios, including requirements changes, to see how it behaves. If necessary, the SE may code the architecture into some sort of analytical tool, plugging in known performance characteristics for some components, parametric estimates for others, and educated guesses for yet others. This provides a baseline for comprehension. The performance data is only as precise as the estimates, but at least the areas with data voids are illuminated and can be targeted for update as better information is found.

Risk Management. A SE who fully comprehends the characteristics of a system can pinpoint likely risk areas and take steps to mitigate the risk. It might be said that the ultimate success of any engineering development effort is directly related to the successful control of risk throughout its development. If the technical, cost, and schedule risk factors associated with each component of the system can be quantified, risk can be managed. Resources can be focused on those aspects of the development which harbor the greatest potential for creating problems later, and preventative measures can be taken to avert these problems. This analysis and decision-making process forms the essence of Systems Engineering.

Isolation of Creativity. It has been long understood that efficient development by large teams requires effective separation of labor. As idyllic as it sounds to have everyone on a team in on every decision, and to grow a culture of people equally good at everything, it is not conducive to competitive product development. A team should have specialists in the various fields that contribute to the product, and the SE must utilize the architecture to harness the diverse capabilities on the team. Just as in the case of the architect of a Gothic cathedral, the SE creates bounded "pools of creativity", within which a specialist is free to ply his trade to the best of his ability, without the fear of impinging negatively on someone else's area. The SE has made the decisions that affect the system in a global sense, and by so doing has empowered the specialist to develop focused solutions. As the cathedral architect has ensured that diverse pieces will come together to create the desired ambiance, so has the SE used the system architecture to guarantee that desired interfaces and overall style are attained.

Communication: One of the authors was asked by a manager long ago what the key attributes of a good SE would be. This particular author, who shall not be named but whose initials are not A. D., floundered about with discussions of requirements allocations and specmanship, etc. ad nauseam, until the manager put him out of his misery with the single word "communication". As time goes on, the manager's conclusion has been validated countless times. But a project cannot be left to the personality type of the Systems Engineer. The manager did not mean that a SE must be gregarious, vocal, nor even clear in his expression. The architecture holds the key to successful communication.

The SE can create simple tools to capture the important aspects of every facet of the architecture, and can require their use by the developers. Examples include component code templates, documentation templates, and guidebooks. If the tools are straightforward and their use contributes to productivity (especially if they handle some of the mundane aspects of design), they will be embraced by developers, part of the communication challenge will be solved, and the enforcement aspect of the SE's job will be easier.

Communication continues between the SEs and the developers throughout the development process. The architecture forms the guide for the SE to devise meaningful metrics that measure progress, indicate adherence to design rules, and illuminate emerging indications of flaws in the architecture.

Architecture also provides the best mode of communication with the maintainers of the system. A maintainer is much more likely to respond rapidly to change during the life cycle if he understands the behavior of the system, where the change must be implemented, and the ramifications of the new implementation on the entire system.

F-22 PROGRAM EXPERIENCE

By now, the reader should understand the fundamental concept of architecture, and its relationship to systems engineering. The following discusses how this concept is being used within the F-22 training system program.

Training System PAT

A Process Action Team (PAT) was formed to develop an overall strategy for the development of an architecture for the training system as a whole. One of the earliest challenges addressed by the PAT was the synthesis of a definition of the term "architecture." An effort was made to develop a standard definition for the term, facilitating communication among members of the PAT. Care was taken to avoid coming up with a "least common denominator" definition, with the argument that too loose a definition would be meaningless as a communication tool.

The architecture PAT consisted of members with widely varying backgrounds. Some members were managers, while others were engineers. Most of the group had maintenance trainer credentials, with fewer having a PTS perspective. Expertise varied from hardware to software to courseware. In the initial discussions, it was difficult to achieve a common level of understanding of the concept of architecture; each PAT member had his own interpretation, based upon his individual background. It became clear that a common definition of architecture would need to be found, otherwise the task of developing an architecture would be impossible.

A strawman definition was proposed by Lockheed-Fort Worth Company (LFWC.) LFWC suggested that the PAT use the working definition of "a framework that conveys engineering decisions regarding the partitioning of a system and establishes the coordination strategy that constrains how the partitions communicate, are controlled, are observed, and are synchronized." The Lockheed definition was the subject of much debate, insofar as not every member of the group was comfortable with the concepts it embodied. It was found to have weak areas, such as the fact that it was recognized as a good definition for software[3], but fell short for hardware, in that it did not address physical constraints or electrical characteristics. It was determined that the definition of "architecture" could not be limited to the software, but must apply to the <u>system</u> as a whole. Also, owing to their different backgrounds, each member had an individual "comfort zone," outside of which he could not concur with the definition. A number of members argued for a more general definition, whereas others favored a more explicit one. It appeared that it would be impossible to settle on common ground.

At that point, it was proposed that a set of options be identified, from which a single definition would be chosen. The approach taken was to generate a series of upwardly-compatible definitions, and step through them one by one until a common "comfort zone" could be found. The intent was to start with a minimalist definition which was acceptable to everyone, and add detail incrementally. When the detail began to fall outside the "comfort zone" of an individual member, an explanation of the added concept might be sufficient to expand that member's acceptability boundary. The following set of definitions was created and discussed:

1. "A philosophy about partitioning a system." An architecture can be considered to portray a view of how a system would be broken up to implement a specific philosophy. For example, a conscious intent to fit the software into a specific hardware configuration will force it to be partitioned in a certain manner, regardless of other factors.

2. "A *framework that depicts* the partitioning of a system." Rather than simply stating that the system will be partitioned to implement a certain philosophy, it shows how the system is actually partitioned to reflect this philosophy.

3. "A framework that depicts the partitioning of a system, *and describes the interfaces among partitions*." This definition adds a level of detail to the preceding one, describing the connectivity among partitions.

4. "A framework that depicts the partitioning of a system, *and establishes* interface *relationships and coordination* among partitions *to a specified level*." This introduces dynamic considerations, through the concept of <u>coordination</u>. This is an important aspect of architectures in general, and particularly so in real-time applications. In addition, this definition adds the idea that a level of detail constraint will be imposed prior to the partitioning.

5. "A framework that *conveys engineering decisions regarding* the partitioning of a system *in accordance with predefined system requirements*, and establishes interface relationships and a coordination *strategy* among partitions to a specified level." This establishes a relationship between

---

[3] This definition was, in fact, adapted from the definition of the structural model defined by the SEI's Structural Modeling Guidebook.

architecture and engineering; the responsiveness of the architecture to system requirements; and the idea of a coordination strategy. The thought here is that it is no longer sufficient to define just the coordination among the known partitions, but to establish an open-ended approach that can support future growth.

6. "A framework that conveys engineering decisions regarding the *logical and physical* partitioning of a system in accordance with predefined system requirements, which establishes *hardware and software* interface relationships among partitions to a specified level, and defines a coordination strategy *that constrains how the partitions communicate, are controlled, are observed, and are synchronized."* This definition adds clarity, and the applicability of the architecture to the real-time domain begins to emerge.

7. "A framework that conveys engineering decisions regarding the logical and physical partitioning of a system in accordance with predefined system requirements, *and establishes and/or invokes standards for adherence by the designers of individual partitions;* it establishes hardware and software interface relationships among partitions to a specified level, and defines a coordination strategy that constrains how the partitions communicate, are controlled, are observed, and are synchronized." The last version adds the concept of standards into the definition, an important element in the establishment of consistency.

Selection of a Definition. Given the seven options above, and following considerable discussion among the members, the PAT ultimately adopted definition #5. It was found that this definition fell within the "comfort zone" of all members, proving that the communication which had taken place during the definition process had served to transmit each member's perspective to the others. An acceptable definition had been found which was not the "lowest common denominator."

PTS Architecture Design Team

ADT Architecture Definition. Following the PAT's selection of a definition, the PTS ADT addressed the architecture issue from its lower-level perspective. Unsurprisingly, the ADT chose a definition which was slightly more complex:

"A framework that conveys engineering decisions regarding the logical and physical partitioning of a system that meets system requirements *and constraints*, which establishes hardware and software interface relationships among partitions to a specified level, and defines a coordination strategy that constrains how the partitions communicate, are controlled, are observed, and are synchronized."

This variant of PAT definition #6 recognizes that there are other constraints levied on a design which are not formally called out as requirements, and may affect its partitioning.

Application of Definition. The PTS ADT went a step further than the PAT in it applying this definition. In the course of developing engineering concepts for the pilot trainers, the ADT identified a number of different approaches to building training devices. Recognizing the need for an architecture, it was determined that further investigation would need to focus on those approaches which represented true architectures. The ADT definition of architecture was thereby used to identify valid architectural options from a list of trainer design approaches, and cull the list for the more thorough investigations to follow.

High school chemistry students are familiar with the Litmus Test, wherein a chemically treated paper is dipped into an unknown liquid to determine whether it is an acid or a base. A chemical reaction will cause blue litmus paper to turn red in the presence of some solutions, allowing those solutions to be positively identified as acids; but the reaction will not occur - and hence the paper will remain blue - when immersed in "something else" (a base or a neutral solution.) In a like fashion, the ADT devised a "litmus test" to ascertain whether a proposed approach could be called an architecture. If the approach met the criteria - caused a reaction, as it were - it would be positively identified as an architecture. If it failed to meet the criteria, it would be labeled "something else."

The criteria against which the proposed approaches would be evaluated were based on the ADT's adopted definition. Essentially, to be evaluated by the ADT, an approach had to address system partitioning, establish inter-partition interfaces, and define a coordination and communication strategy, within the context of the established requirements and constraints. If the approach met all of these criteria, the "reaction" would occur, and the approach could be categorized as an architecture. If not, being deficient in some way, it would need to be either modified or dropped from further consideration.

Evaluation. The ADT generated a list of sixteen potential approaches to the development of pilot training devices, of which ten were identified as passing the "litmus test" for qualification as architectures. (Some of the options were later determined to be subsets of one another. The ten were subsequently collapsed into a set of six orthogonal options.) A few examples of these approaches follow:

An option which qualified as an architecture was the "Domain Architecture for Reuse in Training Systems (DARTS)" [Crispen 93]. Through its inheritance of partitioning and control concepts from the Air Vehicle Structural Model (AVSM)[4] and the Modular Simulator[5] program, DARTS was found to address all of the criteria for further consideration.

---

4 As described in the Structural Modeling Guidebook.

5 The Modular Simulator Program, A.K.A. ModSim or Have Module, was conducted in the late 1980's by the ASC Training System Program Office (ASC/YW) to develop a standard, high-level functional hardware partitioning scheme for training simulators.

Another alternative which passed the "litmus test" was the "Surrogate CIP." This approach attempts to minimize cost by substituting commercial computer hardware for the Common Integrated Processor avionics hardware in the training devices. Being based on the Avionics Integration Laboratory software coordination and partitioning approach, as well as commercial hardware standards, it was found to meet the architectural criteria.

An option which failed the test - i.e., did not qualify as an architecture - was "No Air Vehicle Hardware," an approach intended to build trainers without using any flightworthy avionics, cockpit controls, or display devices. While it might be considered an architecture from a hardware standpoint, this alternative failed the "litmus test" because it defined no scheme for partitioning or controlling software processes.

The remainder of the suggested approaches were assessed in a similar manner, allowing several approaches to be dropped from further consideration.

<u>Relative Quality.</u> Beyond just categorizing an approach as either an architecture or as "something else," a number of factors were used to assess the relative quality of an architecture. These factors[6] are as follows:

Scalability: Support for higher or lower fidelity devices.

Extensibility: Support for the addition of new functionality.

Distribution Across Processors: Allows the use of a multiprocessor hardware architecture, without requiring changes to the architecture.

Manage Complexity: Comprising a relatively low number of different object types; interactions among objects are simplified.

Contain Cost: Includes tools to measure and observe cost factors.

Manage Data Bases: Provides mechanisms to manage and control the amount of information needed to provide effective training.

Concurrency: Supports the rapid incorporation of air vehicle configuration changes into the training devices.

Support Changing User Requirements: Supports the rapid incorporation of changing user requirements into the trainers.

Simulate/Stimulate/Emulate Mix: Allows the deferment of decisions regarding whether specific components of the trainer will be simulated, stimulated, or emulated.

Manage Malfunctions: Supports the future addition of malfunctions, as well as all initial malfunction requirements.

Cost Considerations: Provides the means to measure EMD, Production, and Support costs.

In addition, a good architecture was defined as one which could accurately predict the behavior of the ultimate design; provide an understanding of the overall system; communicate implementation rules to designers; include mechanisms and tools to manage the development; provide quantifiable assets; and control complexity.

As an aid to understanding the nuances associated with each architectural option, a preliminary assessment of the six combined architectures was conducted with respect to these factors. Unsurprisingly, it was found that those architectures which were developed for the training system application were better suited to the F-22 PTS than those developed for the laboratory environment. In the examples above, DARTS fared better than Surrogate CIP for just this reason.

Lessons Learned

<u>PAT Lessons</u>. The lessons which can be taken from the PAT exercise to select a definition of the term "architecture" may be summarized as follows:

1. Since the concept of an architecture can be difficult to grasp, especially by non-technical personnel, a consensus definition needs to be agreed upon which falls within the "comfort zone" of each individual in the group.

2. Architecture can be defined using a series of discrete, additive concepts, to yield a hierarchy which can support the various "comfort zones" without reducing the definition to the lowest common denominator.

3. By presenting the group with a range of definitions, as opposed to a single one which embodies a considerable number of different concepts, each individual can find his own "comfort zone," and identify the specific factors which fall outside this boundary.

<u>ADT Lessons.</u> Additional lessons which were learned from the subsequent ADT activity might be encapsulated as:

1. The level at which an architecture will be applied will influence the consensus definition. As one gets closer to the implementers of the system, the more detailed the "comfort zone" becomes. The ADT "comfort zone" was accordingly lower in the hierarchy than the PAT definition.

2. It is not sufficient to simply state that an approach is or isn't an architecture, and proceed to implement it. All architectures are not equal; there are varying degrees of quality, even within the established set of architectures. To

[6] This list was derived from information provided to the ADT by Mr. Joe Batman of the Software Engineering Institute.

find the optimum solution for any problem domain, the available architectural options need to be weighed against the relative quality criteria for that domain.

Summary. Defining the architecture term was a significant step in the process of actually developing an architecture, because it established the objective for the activities to follow. This paper will now proceed to describe how this definition will be applied to create an architecture for the training system.

ARCHITECTURE IMPLEMENTATION

Creating the Architecture

The current approach of the Architecture Design Team is to lay out, top down, an architecture for what we expect to be the most functional and highest fidelity pilot training device. The architecture will be scalable to adapt to whatever suite of training media is recommended by the instructional analysts. The intent is to provide a common framework applicable not just across the range of pilot training devices, but for some number of maintenance training devices as well. This is one reason for starting with the most complex device in the training system; it is easier to scale down to other applications that don't require the full functionality or fidelity of the complex system, than to scale up to a complex device from the framework developed for a simpler device.

The ADT is considering a two-tier repeating pattern suggested by SEI. One tier comprises a series of components that represent different domains within the simulator device, and the other tier represents a "foundation" that manages the interactions between components, including communications and activation. Within any given component, the two-tier pattern may be repeated. The ADT, along with SEI, has developed a "first layer", which delineates the boundary lines between parts of the training device that have potentially different internal architectures. The problem represented by each component will be examined to determine whether any of the architectures are sufficiently similar to suggest common solutions. Each component will then be developed until its architecture reaches a level of granularity that provides the aforementioned benefits, e.g. cost predictability, comprehension, etc.

The basic method used to develop the architecture within any given domain will be to examine the problem domain it represents, bound and define the problem, and create abstractions of the problem definition in an iterative fashion until patterns emerge that can be captured in the solution.

The resultant solution will be analyzed for resource requirements, both in terms of real time dynamics and development costs. If possible, the architecture will be captured in a tool that will permit more exhaustive characterization of its behavior and platform requirements.

Using the Architecture

In the near term, the architecture will be utilized as a tool to assess various options under consideration by the instructional analysts, as a framework from which to assess the reusability of air vehicle developmental items that have been pursued by the Training System as "planned reuse" (as well as "opportunistic reuse" options), and as a basis from which to conduct a sound make/buy analysis.

Training Media Cost/Benefit Trades. The ISD process is approaching a phase where multiple device suite options will become apparent, and the analysts will require rapid analysis of the costs of various approaches as input to their trade studies. The SEs intend to utilize the architecture as a sort of design database, where the various media options represent different views into the database. This will permit a credible, rapid assessment of the viability and cost of options under consideration. This is one example of how the interaction that is possible between ISD and Systems Engineering can result in an optimal mix of training media for effective pilot and maintainer training.

Reuse Assessment and Recommendation. It is apparent that the reuse of a component from a foreign architecture will strain the target architecture in some manner. To some degree, the attributes that were built into the target architecture will be compromised by the introduction of an entity whose structure represents a different pattern set. Drawing again on the analogy of the architecture of buildings, [Alexander 77] states that in short, "no pattern is an isolated entity. Each pattern can exist in the world, only to the extent that it is supported by other patterns: the larger patterns in which it is embedded, the patterns of the same size that surround it, and the smaller patterns which are embedded in it." When this principle was realized by the F-22 PTS, an effort was made to anticipate the ultimate architecture of the training devices, and use its patterns to influence the structure of certain items targeted for potential reuse. Considerable success was achieved in the areas of Flight Dynamics Simulation, Air Data Sensor and Gyro & Accelerometer Simulations, Electronic Warfare (EW) RF Sensor Simulation, EW Countermeasure Controller Simulation, EW Missile Launch Detector Simulation, Comm/Nav/Ident (CNI) Radio, In-Flight Data Link, TACAN, ILS, and IFF Simulations. All of these simulations are being developed by air vehicle IPTs, and each IPT adopted a close variant of the AVSM (with a little encouragement, as well as SEI support, from the Training System IPT.) In any case, every reuse candidate must be analyzed with respect to the strain it places on the architecture (which impacts directly on life-cycle cost) versus the development cost of an alternative.

Make/Buy Analysis. As much as the architecture can serve as a tool for comprehension and communication to developers, it has utility to provide industry with a clear definition of the problem; every bit as clear as that available to the prime contractor, ideally. The F-22 training system is pursuing means by which industry can participate in the architectural development process with domain experts, in an attempt to build the best solution for the user that industry can create, and at the same time get industry familiar with the various problems being faced on the F-22 program. This will enhance the likelihood that a set of well-positioned contractors can be selected to build various parts of the F-22 training system. The architecture will also be used to provide insight into exactly what components, if any, are best built by the prime, while ensuring that pieces built by different companies will integrate well.

None of this discussion precludes a total buy option. Whether the device is built in-house or by a vendor, the problem should be well understood. The F-22 Training System is sensitive to the fact that our problems are of a sufficiently unique nature that vendors are not likely to desire involvement without a solid understanding of what they're getting into. The problem does not necessarily scale well from previous fighter simulation programs, so the existence of a descriptive architecture is a key to its success.

Managing the Development

Once the architecture is established, the SEs will create tools which will enhance their capacity to enforce the architecture. For example, component packages will be templated to ensure consistent treatment of interfaces and coherent style. Documentation standards will be published and perhaps templated to assist in the creation of a design database that will be used to analyze the ongoing development and define the executive layer. A package will be developed to permit rapid orientation of new developers to the context within which they will operate, and the specific requirements for the components they will be responsible for.

The SEs will monitor the development process using the standard documentation produced by the developers. The data being provided will be used to update the parameters that were assumed in the analysis of the architecture to validate expected system behavior. The iterative analysis will pinpoint emerging risk areas, which will be tended to quickly by reallocation of resources or by modification of the architecture. The SEs will resolve ambiguities and enforce adherence to the system design. This process will continue throughout integration and test.

SUMMARY

The role of an architecture within the Systems Engineering process is difficult to express, as few people agree on the definition of architecture and few companies have a strong Systems Engineering process.

The F-22 Training System wrestled with these issues and came to consensus on a range of complimentary definitions for the concept of architecture, and has considered how the architecture will be used by the SEs to confront the large problem ahead.

<u>Applicability of F-22 Lessons.</u> The F-22 Training System must provide training for an aircraft that is highly integrated, flexible in mission, smart enough to fuse data before the pilot sees it while offering the pilot pre-fused information on request, and reconfigurable in real time to adapt to failure conditions. Past "black box" simulation techniques will never capture the myriad possible modes. More recent trends to create strong mapping to air vehicle "boxes" won't work in many cases because we are not dealing with a federated system, and air vehicle software is not tied to any particular air vehicle processor; functions may move around.

What is occurring on F-22 is simply an example of systems scaling to levels of complexity that cannot be addressed with techniques that worked for smaller systems. Industry found that the last time there was a quantum leap in aircraft system complexity, the AVSM architecture was adequate to allow the aircrew training systems to keep pace. Now that another scaling leap is occurring with the F-22, it appears that the application of strong Systems Engineering throughout development, fully leveraging an architecture exhibiting the same principles as the AVSM, will prove capable of solving the F-22 Training System problems (and not just for aircrew training this time). It seems reasonable to expect that the process described can provide a generic paradigm with applicability to many domains facing similar problems of scaling complexity.

ACKNOWLEDGMENTS

The authors wish to acknowledge Mr. Joe Batman of the SEI, for his contributions toward the synthesis of an architecture for the F-22 PTS; Mr. Michael Rissman of the SEI and Mr. Bill Schelker of ASC, for their seminal efforts in the development of architectural constructs for training devices; and the members of the F-22 Training System Architecture PAT and the PTS Architecture Design Team, for their efforts leading to the findings described herein.

BIBLIOGRAPHY

Alexander, C., "A Pattern Language," Oxford University Press, 1977

Crispen, R.G., Freemon, B.W., King, K.C., and Tucker, W.V, "DARTS: A Domain Architecture for Reuse in Training Systems," 15th I/ITSEC, Nov '93.

Petroski, H., "Design Paradigms," Cambridge University Press, 1994.

"Structural Modeling Guidebook (Draft)," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May '93.

"Systems Engineering Management Guide," Defense Systems Management College, Ft. Belvoir, VA, Jan '90.

LESSONS LEARNED IN DEVELOPING MULTIUSE SIMULATION FOR F-22

Dorothy M. Baldwin, James H. Gault, and Stephen S. Zimmel
Lockheed Fort Worth Company (LFWC)
Fort Worth, Texas

ABSTRACT

Multiuse Simulations are even more critical in light of current budget constraints. Early planning during F-22 development has provided a unique opportunity to maximize simulation synergism across an entire Weapon System. Via Integrated Product Teams (IPTs), the Air Vehicle, the Support System, and the Training System are being developed concurrently. Potential simulations for REUSE by the Training System were identified early to be able to incorporate training requirements into Air Vehicle and Engineering lab developments.

This paper describes "Lessons Learned" in developing simulations to satisfy multiple engineering laboratory and training requirements and also provides examples of specific cases where Training System personnel have acted as "integrators" between various Air Vehicle IPTs.

A good example is the development of the Flight Dynamics Simulation (FDS). FDS has completed Preliminary Design Review (PDR), Critical Design Review (CDR), coding, integration and testing, and will be operational in the Vehicle Management System (VMS) Integration Facility (a full-up pilot-in-the-loop engineering flight simulator) by the time this paper is presented. All potential users, including training system personnel, were involved in requirements, review, and approval cycles. All identified training requirements have been met. Examples are given of how FDS development "Lessons Learned" have been shared with other REUSE engineering simulation developers.

Challenges that lie ahead and the processes being put in place include (1) how to develop a robust, flexible design based on early requirements that we know will change, i.e., how to incorporate REUSE simulations into the final media that result from Instructional System Development (ISD) and provide these REUSE simulations to the ultimate training simulator designer and integrator and (2) how to update the REUSE simulations during the Weapon System life-cycle while satisfying the requirements of diverse users.

ABOUT THE AUTHORS

Dorothy M. Baldwin is an Engineering Project Manager at Lockheed Fort Worth Company, currently assigned as the Integrated Product Team (IPT) Manager for the Lockheed portion of the Pilot Training System Devices for the F-22 aircraft. During her ten years at Lockheed/General Dynamics, Ms. Baldwin has led efforts related to foreign customer initiatives in both training and engineering laboratory development; has provided technical and management leadership for a multimillion dollar engineering flight simulator laboratory development; and has led a tank simulation development in support of General Dynamics Land Systems Division. Prior to coming to General Dynamics, Ms. Baldwin spent seven years as a physicist working for the Naval Training Equipment Center (where she led multiagency and industry efforts in sensor and visual simulation technology development) and spent the previous twelve years teaching physics and developing related courseware and curriculum at a number of universities, colleges, and high schools. Ms. Baldwin has a BA from Hartwick College and an MA in physics from Kent State University and is a certified physics teacher in two states.

James H. Gault is an Engineering Specialist Senior at Lockheed Fort Worth Company assigned as Software Product Manager for the F-22 Training Systems group. His current responsibility involves leading an effort to develop a common F-22 flight dynamics simulation for reuse in several engineering laboratories and Pilot Training System devices. Mr. Gault's experience, prior to the F-22 Program, included four years developing engineering and simulation software for AFTI and other F-16 derivatives and over ten years developing real-time software for F-16 avionics test equipment at LFWC. Mr. Gault holds a BS in both chemistry and mathematics from the University of Texas, Arlington, and the University of Houston, respectively.

Stephen S. Zimmel is an Engineering Specialist at Lockheed Fort Worth Company's flight simulation laboratory, where he has been employed as a hardware and systems design engineer since 1983. He is currently responsible for the development of the LFWC F-22 flight simulators being used in the analysis and testing of the F-22 flight controls and vehicle management systems. Mr. Zimmel holds a BS degree in electrical engineering from Pennsylvania State University.

LESSONS LEARNED IN DEVELOPING MULTIUSE SIMULATION FOR F-22

Dorothy M. Baldwin, James G. Gault, and Stephen S. Zimmel
Lockheed Fort Worth Company (LFWC)
Fort Worth, Texas

BACKGROUND

Multiuser (or) REUSE Simulations are becoming even more critical in light of current budget constraints. Early planning during the F-22 Program has provided a unique opportunity to maximize simulation synergism across an entire Weapon System. Via Integrated Product Teams (IPTs) during the Engineering and Manufacturing Development (E&MD) phase[1], the Air Vehicle (A/V), the Support System (SS), and the Training System (TS) are being developed concurrently.

REUSE is an integral part of the F-22 program and F-22 contract; in fact, it is part of the F-22 contractual award fee criteria. References to REUSE appear in a number of F-22 documents, including the Integrated Master Plan and the Integrated Master Schedule (Reference 1), and both the Weapon System (Reference 2), and the Training System Specifications (Reference 3). The F-22 Weapon System Software Development Plan (WS SDP) (Reference 4) and the Training System Software Development Plan (TS SDP) (Reference 5) define the REUSE process.

Air Vehicle IPTs are developing portions of the air vehicle simulation for incorporation into the Pilot Training System devices. Potential reused components for the training system were defined and worked early in the E&MD program to enable their inclusion in air vehicle and engineering lab developments. In parallel with efforts to maximize REUSE, a rigorous Instructional Systems Development (ISD) process is being conducted to define the total training system

The REUSE effort described herein is actually one example of a much larger movement in the industry and at LFWC to improve software development processes. Lockheed is a member of the Software Productivity Consortium formed by aerospace companies for this purpose in 1985. One product of the consortium, Ada-Based Design Approach for Real-Time Systems (ADARTS), was adopted for the F-22 program. In addition, LFWC was recently evaluated by the Software Engineering Institute (SEI) via their Capability Maturity Model as a Level III. Level III certification means, "The process for engineering and management is documented, standardized, and integrated into an organization-wide software process." The Flight Dynamics Simulation (FDS), described herein, was one of the modules tracked for compliance. The Defense Department expects these new processes to save billions of dollars (Reference 6).

A follow-on to a 1991 paper by Baldwin and Landry (Reference 7), this paper describes lessons learned to date in the development of unique processes to allow simulations to satisfy multiple Engineering Laboratory Development and Pilot Training System device requirements. This paper concentrates on the successes encountered and challenges overcome (and yet to be overcome) for LFWC Pilot Training System Devices (PTSD) Software REUSE items. Though REUSE is being addressed in other areas of the F-22 Weapon System, this paper is limited to the areas cited above.

INTRODUCTION

REUSE goals include improved quality, supportability, potential cost reduction or cost avoidance, and potential schedule savings or schedule delay avoidance. The TS SDP (Reference 5) addresses all aspects of Training System Software development through the Weapon System life cycle. The Training System will do a Make vs. Buy for entire systems and applicable subsystems. Any potential pilot simulators, identified by ISD, could be bought from vendors. Applicable REUSE items (that survive the Make vs. Buy process) could then be provided as contractor-furnished equipment (CFE). An F-22 "Joint Procedure" (Reference 8) provides guidelines for implementing REUSE across the entire team. Three types of REUSE are identified in Reference 8: Planned REUSE, Opportunistic REUSE, and Anticipated REUSE.

Planned REUSE – IPTs identify common assets within or across IPT boundaries and enter into a partnership to consolidate commonality to the extent that one IPT becomes a developer and one or more IPTs become reusers.

1 Starting in 1991, E&MD was lead by Lockheed Aeronautical Systems Company (LASC) teamed with Boeing and Lockheed Fort Worth Company (formerly General Dynamics, Fort Worth Division). Boeing is team lead for the Training System.

Opportunistic REUSE – IPTs reuse existing assets and modify them to fit their application.

Anticipated REUSE – The principle of engineering all assets with reuse characteristics to enhance reusability on future programs.

This paper and the previous paper (Reference 1) deal primarily with examples of planned REUSE. REUSE includes design, code, documents, test, data, tools, etc.

DEVELOPING REUSE SIMULATIONS USING THE INTEGRATED PRODUCT TEAM (IPT) CONCEPT

One IPT is assigned responsibility to meet REUSE requirements of several IPTs, with potential reusing IPTs invited to participate in all phases of development and requirements definition through reviews, software product evaluations (SPEs), and testing. Plans are established for including all potential reusers in the software change process through the life cycle. Figure 1 is a summary of planned reuse for the LFWC PTSD responsible areas, including all potential reusers.

	User/Laboratory							
Simulation	VIF	VSS	SDL (1)	SDL (2)	AIL	SIL	FMS	PTSD**
FDS*	✔	✔			✔	✔	✔	✔
CNI			✔		✔	✔		✔
EW			✔		✔	✔		✔
IRS	✔	✔	✔	✔	✔	✔		✔
VMS/U&S			✔	✔				
SMS			✔	✔				

*Development used as case study in this paper.
**Final REUSE items subject to results of ISD and trade studies.

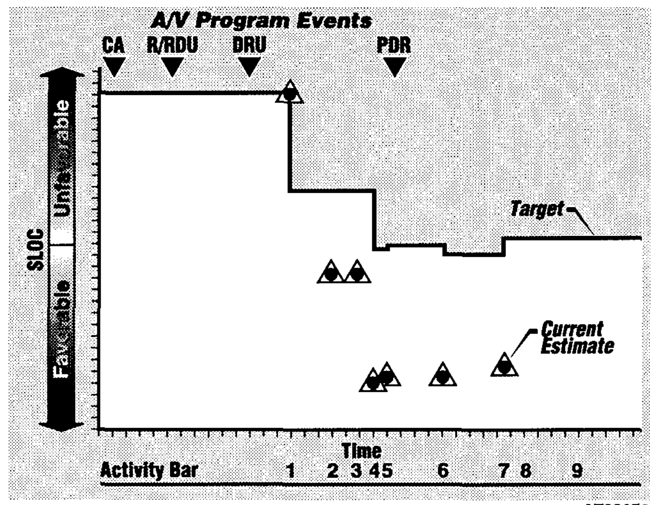
AT02149

Figure 1 F-22 Simulation Planned REUSE

Preliminary PTSD requirements were identified in a timely manner through the use of "quick looks," which consisted of preliminary ISD analysis of critical REUSE areas. Contractual tasks were created to identify training requirements for engineering lab simulations. The PTSD IPT leader had approval rights of the software documents.

We have a program target that PTSD requirements cannot impact engineering simulations more than 20 percent. Technical Performance Metrics (TPM) are used to measure how well we are doing with respect to our target. This TPM (Figure 2) is used to measure the simulation software Source Lines of Code (SLOC) for LFWC-responsible simulations required to support PTSD unique requirements.

The target value and the current estimate are recalculated at each program event along the activity bar. Program events numbered along the activity bar are major reviews for the LFWC PTSD REUSE simulations. Figure 2 shows performance better than the target and, therefore, in the favorable area.



AT02150

Figure 2 LFWC Simulation Software SLOC Allocated to PTSD Unique Requirements

MULTIUSE SIMULATION DEVELOPMENT – A CASE STUDY

The Role of the Developing Laboratory – The Flight Dynamics Simulation (FDS) will be used to illustrate the progress made on the MULTIUSE (or REUSE) simulation development.

The Flight Dynamics Simulation (FDS) Computer System Configuration Item (CSCI), intended to meet high-fidelity F-22 Airframe Simulation requirements of engineering labs and training systems, was developed as a crossflow item to support the six users shown in the shaded portion of Figure 1. The simulation was originally developed for the VMS Integration Facility (VIF) . Because of PTSD deliverability, FDS was required to meet F-22 deliverable standards as specified in the WS and TS SDPs, i.e., to be DoD-STD-2167A, -2168, and MIL STD 1803 compliant and to be written in Ada. The FDS was co-developed by Flight Simulation Laboratory (FSL) and PTSD personnel. The FDS CSCI has been designed with the knowledge that extensive REUSE will occur. All designs, code, documentation, and test procedures will be available for REUSE by the Engineering labs and PTS.

To put the FDS into context, a brief overview of the VIF is provided. The architecture diagram for the engineering development laboratory (Figure 3) illustrates the relationships between the various hardware and software elements that comprise the system.

The software components are shown as circles with the hardware components and their interfaces represented by rectangles.

At the core of the software products are the Flight Dynamics and Vehicle Management System Sensor simulations, that were designed to crossflow, or for

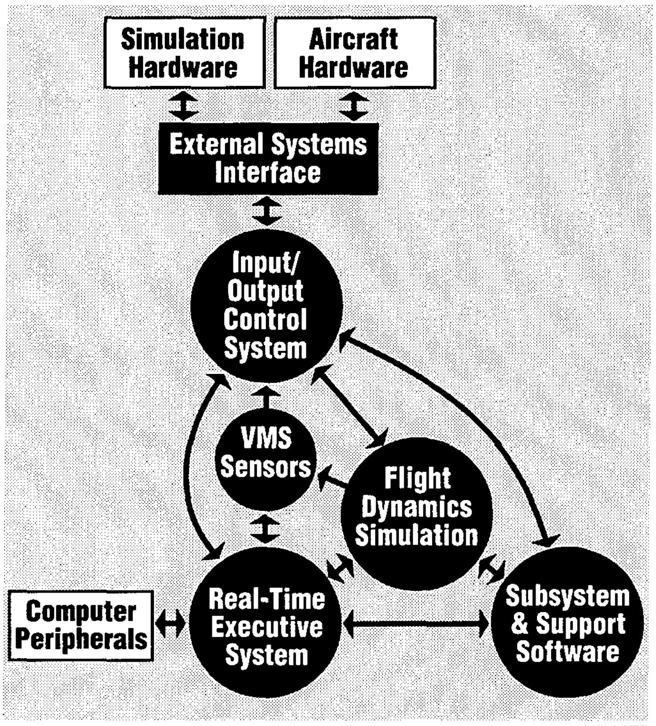


Figure 3 Engineering Development Laboratory System

REUSE, into the Training Simulators. The FDS is the model of the physical motion of the airframe, excluding (to the greatest extent possible) the internal aircraft systems. The VMS Sensor simulation models the behavior of the F-22 Vehicle Management System's unique set of gyros and accelerometers.

The remaining software includes the real-time executive that provides the user interface and overall simulation control function; the input and output control software (which scales, gathers, and scatters data between the simulations and the external hardware systems); and the subsystems and support software (which provides simple models of the other aircraft systems necessary to satisfy interfaces not provided by actual aircraft hardware). A potential layout for the PTSD simulator is described in Reference 7.

Reuse Strategies for FDS – Strategies used to meet requirements of several IPTs include:

- Inviting potential reusing labs and IPTs to participate in review and software product evaluation (SPEs). Plans were to include all potential reusers in the software change process.
- Partitioning the FDS to be as independent of a specific computer as possible through the use of structural modeling techniques.
- Avoiding constant revisions resulting from aircraft interface changes by modeling only physics – not avionics or aircraft subsystems.
- Structuring the software so that it requires only a tailored shell to handle system calls, i.e., no input or output software is required.
- Accommodating different update rates, i.e., programmable Δt.

Role xof Pilot Training Systems – PTSD involvement began in the concepts definition phase and was part of the E&MD proposal. Involvement continued during the requirements definition phase with the review and submission of requirements. During the preliminary and critical design phases , PTSD was a contributor to the Software Product Evaluation (SPE) process. SPEs are required by DoD-STD-2167A to be performed on deliverable products during the software development phases.

PTSD had an integral role during the coding and integration phases by having a number of PTSD software engineers included as part of the FDS development team. Although this was unique among the REUSERs, it was considered to be critically important because of the different philosophies governing the engineering and training simulations.

The role of PTSD during testing of a REUSE item is similar to that of the earlier phases; PTSD participated in the SPE of all FDS test procedures. PTSD IPT members are working within the PTSD IPT (which includes SPO) and with the A/V IPT to come up with tests that will satisfy trainer deliveries and engineering lab developments, through the life cycle. This is ultimately the only way to achieve true synergy between PTSD and the engineering lab developer, and the resulting cost savings from REUSE. The goal is that (throughout the life cycle) once a change is successfully tested in the engineering lab, a copy of the software can be shipped to the trainer – for immediate concurrency between the engineering lab simulator, the updated airplane, and the training simulator.

Phase-by-Phase Results Summary and Lessons Learned – A Case Study

Requirements Definition Phase – During the requirements definition phase, the Software Requirements Specification (SRS) and Interface Requirements Specification (IRS) were developed, SPEed, and released after the Software Specification Review (SSR). All requirements were traced to three areas (Figure 4), which presented a difficult challenge, especially when requirements were fluid and would most likely remain fluid for some time. This influenced our decision to go to the structural model approach described later.

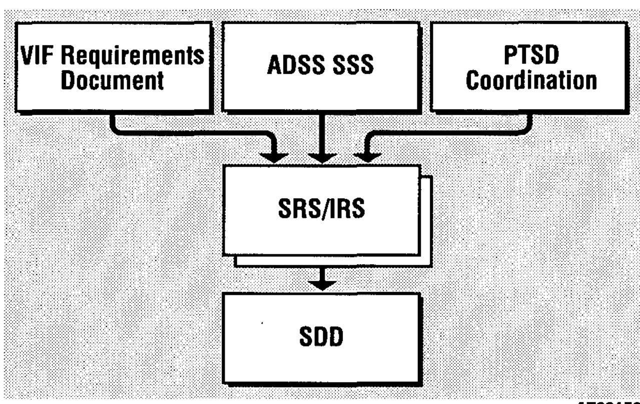


Figure 4 F-22 Simulation Requirements Traceablity

As was touched on earlier, the FDS was functionally partitioned for maximum reuse. FDS provides airframe dynamics, ground handling reactions, aerodynamics forces and moments, mass properties, and atmosphere and wind models. FDS does not provide control and monitoring features, propulsion, actuator, airframe sensors, and flight control models, or aircraft hardware interfaces.

A great deal of confusion resulted from our choice of software product names. Users tended to have entirely different functionality expectations based on their local cultures and previous experiences. For example, the term "flight dynamics" may mean an entire flying airframe – including aerodynamics, flight controls and a propulsion system – to some, but simply equations of motion to others. We would encourage REUSE suppliers to communicate the limitations of the REUSE software as well as the benefits so that potential users are not lulled into the false belief that a single CSCI contains more capabilities than the supplier intends to provide. This should reduce the number of change requests that arise as development proceeds through the design phases and should thereby reduce the likelihood of costly additional development. This clear definition of the boundaries of the REUSE item should also enhance the possibility of potential REUSE on future programs, i.e., "Anticipated REUSE."

A software architecture, consistent with SEI's air vehicle structural model (AVSM) (References 9-12) was developed to aid in understandability, maintainability, extendibility, ease of rehost, ease of adding and modifying malfunctions, and scalability. This architecture is an object-based design with constraints on program communication and coordination. A design decision was made to implement faults at the lowest logical level. Benefits of this structural model architecture include consistent interfaces, i.e., no surprises; proven concept because of common industry use; and easy accommodations for future growth.

In our efforts to "tie down" diverse requirements, we may have gotten carried away. Based on comments by various people at the first walk through of our requirements documents, we added great detail about the system, especially the interfaces, and crossed the fine line between requirements definition and design. Many of our internal interfaces were defined at the CSCI and this presented a problem in two areas, i.e., maintenance and testability. Many times we had to change the SRS and IRS, not due to requirements changes but because of design changes, such as interfaces, which were documented in the SRS or IRS. We also found that many of these "requirements" were not easy to test at the CSCI level because they were really internal. In the future, we should be more careful about what details are included in the requirements documents.

We should also have been more rigorous in establishing only firm requirements, despite our conviction that the software design be reasonably able to accommodate changes. This might seem to be an obvious recommendation, but it is clouded in this case by the wide temporal gap between the initial development and eventual release to the REUSER. For example, the engineering simulation development must necessarily lead the definition of training requirements – the Instructional System Development (ISD) process - by a considerable time. There seemed to be a tendency for the REUSE customers to get caught up in their involvement in the design process and push for the premature inclusion of requirements based on their perception of the future design. A barrier should be maintained between allowing reusers to guide the development toward reusability and allowing them to establish false requirements as firm ones. It is much cheaper in the long run to design a system which can easily accommodate firm future changes than it is to continually redesign the system based on a set of dynamic current requirements.

Since the idea of multiuse simulations is new to most software development teams, it is important at this early stage to define the general configuration management concept which will be used throughout the program life cycle (development, production, and support). The process should not be so burdensome as to preclude the efficient rapid prototyping activities that will be required for the engineering simulator's initial support of the weapon system's dynamic design and integration phases. Conversely, the process must maintain sufficient control of the changes to provide traceability and allow the preparation of detailed formal release documents required for the training system. The apparent disparity between the engineering simulator's requirement for rapid change response and the trainer's requirement for rigorous process control were a primary source of conflict in the FDS development. This became know as the "Rapid Prototype Development Problem" and is illustrated in Figure 5.

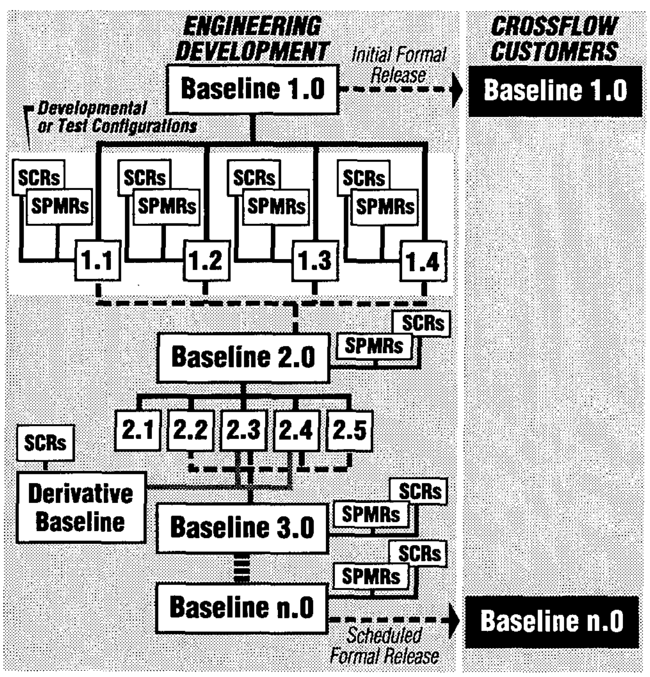


Figure 5 The Rapid Prototyping Development Problem

As is obvious from the illustration, it is frequently the case in an engineering simulator that several versions of the baseline software package may be in use and under configuration management simultaneously. [Note that in the LFWC flight simulation laboratory, two change mechanisms are currently in use. The Simulation Product Modification Request (SPMR) is provided to developers by the end user as an advance notice of changes to the fundamental simulation requirements and typically affects all test configurations simultaneously. The System Change Request (SCR) may be created by either the end user or the simulation developers to make a global or limited change but must be present before a change to the baseline can occur.] The need for simultaneous availability of different test configurations arises because different engineering users need to evaluate the system's behavior based on their unique change or set of changes to the baseline during the same release cycle, often during the same day. Figure 1 shows 1.1, 1.2. 1.3, and 1.4 as user-unique changes that may, or may not, be incorporated into the next formal baseline, 2.0. This is in diametric opposition to the training simulator's situation, where it is not only desirable but mandatory that each trainer of the same A/V configuration, operate with a common software version, regardless of the particular scenario under which the trainer may operate during a given training session. Test configurations that incorporate the individual changes are tracked throughout a given baseline release cycle but are archived at the time of a new baseline release. A new system baseline is defined and released at the discretion of the end user. Figure 1 shows baseline 1.0 and n.0 as formal releases. It is usually based on a related Weapon System milestone (e.g., PDR, CDR) or a major release of some component of the Operational Flight Program (OFP). This release may incorporate any or all of the changes implemented in the test configurations. The entire release-test-release process may be repeated several times prior to a formal release of the software to the REUSE customers. Figure 1 shows baseline 2.0, 3.0, ... n.0-1 with no formal release. With the large number of changes being made in rapid succession, it is obvious that early and thoughtful design of the change control process is very important. (Our method for addressing this problem for crossflow software is described in more detail later in this paper).

By seeing the overall concept defined early, the software developers can become comfortable with the process and gain ownership in it during its refinement in later phases.

Design Phase – Conceptually, the decision to save program resources by REUSE is very attractive, but, as is often the case in the real world, we discovered that the "devil is in the details." Risk Management items and plans were developed early and were constantly modified.

Satisfying requirements of multiple users was on the top of the identified risks list. This proved to be a correct assessment. Common links with the VIF executive software and other F-22 high-end operating systems (HOS) were identified as a REUSE risk. Our risk abatement approach was to minimize links and coordinate with HOS developers. Another risk item was the need to rehost FDS on different platforms. The obvious approach was to design FDS for least dependence upon a hardware configuration. This was accomplished by insulating the FDS from the host computer using the system's executive software and by establishing a common data interface for all applications that was consistent with the structural model.

One of our most significant and painful lessons occurred during the preliminary design phase. All REUSE software had originally been designated as CSCIs for the original developing teams, which, in all cases, were Air Vehicle IPTs. This resulted in the Air Vehicle team having to meet PDR type requirements at Air Vehicle PDR for simulations that did not need to be at that point in their development to meet any IPT's needs. To resolve this, REUSE simulations were designed as non-CSCIs (but developed to deliverable standards) to allow the flexibility to meet schedules and requirements of diverse users. This required that a unique CM process be developed, as described later. We did have sign off on all documents by affected IPT Managers through the development process. This promoted ownership and cooperation but required a willingness to compromise for the overall good of the program.

During the preliminary design phase, developers should be encouraged to resist any temptation to immediately isolate the various pieces of software as independent entities and dismiss external interfaces as unimportant. The external interface names should, if at all possible, be identified early and carried consistently throughout all development phases to ensure that costly discrepancies do not arise during integration.

Between PDR and CDR many changes occurred. Of seventy-eight SRS Requirements, twenty-two were modified, five new requirements were added, thirteen requirements were deleted, the other requirements were unchanged.

Implementation, Integration and Testing Phases – It was during this phase that the benefits of the structural modeling concept and the software architecture chosen during the design phase were demonstrated. Many changes to software were made in an attempt to meet the stringent timing requirements, e.g., the migration of the FDS from two to three and then four processors. The developers believe the chosen architecture supported the rapid restructure and reallocation of this software.

Extensive testing at unit and CSCI levels required considerable coordination with other IPTs for test data. Estimates of the time and effort necessary for this phase were too low because previous simulations had not been tested so thoroughly. This high level of confidence in the simulation was required because the FDS will be used to qualify a safety-of-flight OFP. The proper operation of the real-time simulation was verified by comparison of the output state variable vs. data from the airframe trim, linearization, and simulation (ATLAS) model. Time history comparisons were made by overplotting the data.

CSCI installation, maintenance, and control during FQT were the responsibility of the crossflow software Product Configuration Management System (PCMS) administrator. Tools were provided by F-22 System/ Software Engineering Environment (S/SEE). This worked well.

SHARING OF LESSONS LEARNED WITH OTHER F-22 AIR VEHICLE IPTS DEVELOPING MULTIUSE SIMULATIONS

The VMS IPT, which developed the FDS, has the earliest schedule on the F-22 program for development of its simulations. Many of the lessons learned from FDS development are very useful to other simulation developers on the program. PTSD IPT must cross a multitude of A/V IPT boundaries, thus placing PTSD IPT in a unique position to understand REUSE simulations being developed by various labs (including vendors). The PTSD IPT also has a vested interest in seeing that the REUSE simulations have architectural characteristics that would allow their use in some, yet to be finalized, pilot training device architecture. LFWC's PTSD took the initiative and brought together SEI and FDS developers with vendors who had the responsibility of developing Electronic Warfare Simulation System (EWSS) and later Communication, Navigation, and Identification Simulation System (CNISS). The purpose of these meetings (which took place over several months time for each REUSE item) was to offer the benefits of the experience gained by the FDS development. FDS developers had, under the guidance of SEI, reused a design concept developed under previous trainer programs, i.e., ASVP, C-17, B-2, SOF ATS, etc., (References 9 through 12).

The meetings were very successful, with both the EWSS team and the CNISS team adopting the SEI's structural modeling approach, because it made sense to them. Items shared with the EWSS and CNISS teams that were used for FDS include architecture, specification form templates, code templates, Software Design Document format and words, components, and code. CNISS and EWSS developers reused the specification form templates and the basic concepts. This is potentially a large cost avoidance for the program. Instead of paying for three completely separate developments, much was shared, i.e., the ultimate REUSE and WS IPT in action! The positive for PTS, is that these simulations are now much more attractive for REUSE in a PTSD trainer.

STATUS OF PROCESS DEVELOPMENT FOR ADDRESSING FUTURE CHALLENGES

Providing REUSE Items to Training Simulation Designer – We are faced with the challenge of how to develop a robust, flexible design based on early requirements that we know will change. A further challenge is how to provide resultant REUSE simulations to the ultimate training simulator designer and integrator. We determined that the approach most likely to succeed was structural modeling and PTSD had such a large stake in the outcome that we needed to be proactive.

Structural modeling advantages include ease in changing or adding features without creating a rippling affect. Structural modeling is more understandable, easier to maintain, its subsystems and components are reusable, it eases documentation, and it allows natural propagation of simulated malfunctions. Lessons learned to date, described earlier, support the decision to use the structural modeling approach.

Updating REUSE Simulations During Weapon System Life Cycle – This section only addresses the case study (FDS) described earlier. As stated above in the Rapid Prototyping Development Problem, the requirements of a change process for crossflow software presented many challenges :

- Quick turnaround to support engineering lab activities
- A more controlled release system to ensure commonality across all using communities
- Representation of all users and reusers needs.

A two-tiered change control process (Engineering release and formal release), as depicted in Figure 6, has been set up to address all the above issues.

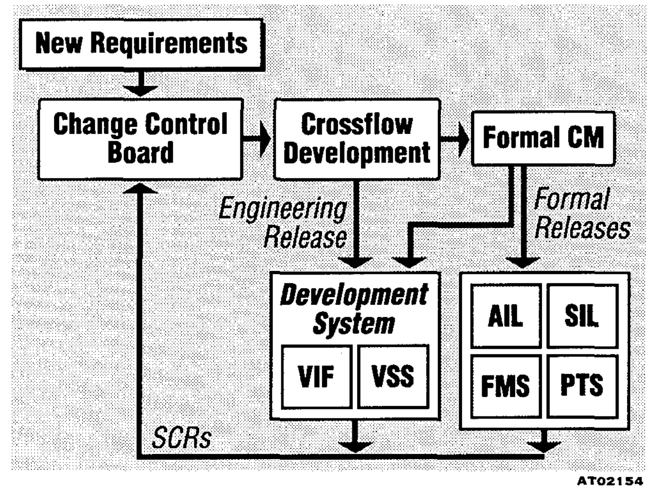


Figure 6 Crossflow Software Change Process

Provisions for an "engineering release" of the software have been made to allow rapid, informal releases. All changes to the baselined software will be tracked via SCRs even within the informal, engineering release process. These releases, while still controlled by the formal CM tool, PCMS, can be generated quickly by eliminating documentation and regression testing requirements at this step. It should be understood that these releases will never be used in support of any formal testing activity in the lab.

Several changes (SCRs) will build up during the engineering release process and those will be grouped together for a block release at appropriate times such as avionics blocks or updated aero data set releases. This process constitutes a more "formal release" to satisfy deliverable requirements. An engineering review board will by tasked with approving all changes that will be incorporated in a particular block release version. Each using lab will be represented directly or indirectly on this board. Each formal release will undergo a regression test appropriate for the types of changes included. Documentation, including a version description document (VDD), will be updated to be representative of that version. Documentation updates will be possible by making use of the SCRs created during the engineering releases. No lab will be forced to immediately update to the newly released version if circumstances dictate, but are encouraged to do so at the earliest opportune moment.

All using labs will be able to submit SCRs to request corrections to problems or to add new requirements they deem necessary. New requirements to accommodate air vehicle changes, new aero data sets or changes due to flight tests may also be input. These new requirements or anomaly corrections must be approved by the engineering review board representing all using labs. The change process will be kept at this level unless disagreements between different labs occur. Any such disagreements would then be elevated up the IPT chain.

CONCLUSION

There are many challenges facing REUSE development, including identifying PTSD requirements in time to achieve maximum synergism across the weapon system. Meeting deliverable training system standards and REUSE requirements of several IPTs (labs) is difficult because for the so many different uses for the simulation, e.g., test and verification, integration, analysis and demonstration, and training. Provisions had to be made for at least two different target computers. And a specific training system concern is for REUSE items to meet vigorous life-cycle support requirements.

In addition to the lessons learned during each phase of development described above, some general lessons need to be mentioned. All lessons are preliminary and will be until the ultimate test of the REUSE items, i.e., integration and life-cycle support in all the target REUSE areas.

We applied the ADARTS process but many of the developers doubt the benefit of this process to our specific application. It is our opinion that ADARTS is better suited for event driven or I/O intensive applications. This is not the case of FDS. In addition, we reused algorithms (if not code) which, along with our performance constraints, dictated a design. ADARTS could not add anything to that structure.

One benefit of ADARTS was that it was used to confirm the architecture and structure we had already established. If ADARTS had indicated something drastically different, we would have reexamined our design.

When you change your "mindset" to include REUSE, you potentially find more than you thought you would. Several ATLAS test cases have been identified for REUSE from the Flight Dynamics and Weapons Separation Group. These test cases will be developed for verification use in the Handling Qualities Simulator, VIF, VSS, and PTSD simulator. In addition, we have begun to look at the IRSS in the VIF as a potential candidate for Opportunistic REUSE.

Chances of success of planned REUSE are enhanced by explicit wording in the contract and in the SOW, early involvement by all potential reusers, and co-development. Defining a software architecture up front improves satisfaction of non-functional requirements, i.e., maintainability, modifiability, and scalability.

Co-development of FDS had its challenges. For example, merging engineering lab culture (quick turnaround; support of flight test; and informal, in-house documentation); with training device culture (deliverable, formal documentation, life-cycle support) was not easy. However, we discovered advantages to this merge. There was a merging of varied skills and expertise, i.e., Ada and deliverable processes expertise from PTSD; flight dynamics and extensive real-time experience from FSL. The result was that the "whole was stronger than the sum of the individual parts."

We believe that the F-22 is developing REUSE strategies that will ultimately result in deploying and maintaining concurrent trainers, at a reduced life-cycle cost.

REFERENCES

1. F-22 Integrated Master Plan, 2 January 1991

2. F-22 Weapon System Specification. RFP No. F33657-91-R-0006, 2 January 1991

3. F-22 Training System Specification, 2 January 1991.

4. Lockheed Aeronautical Systems Company, F-22 WS SDP for E&MD. Contract No. F33657-91-C-0006, November 4, 1993.

5. Boeing Defense and Space Group, F-22 Training System Software Development Plan for E&MD, January 1993.

6. Nordwall, Bruce. "Defense Department Expects New Strategy for Improving Software to Save Billions," Aviation Week & Space Technology, June 22, 1992.

7. Baldwin and Landry, "F-22 Innovations for Concurrent Development of Pilot Training System Devices," 14th I/ITSEC, 1993.

8. F-22 Joint Procedure No. 169. "Software Reuse Guidelines, July 15, 1993.

9. An Object-Oriented Design Paradigm for Flight Simulation, 2nd Edition Technical Report, CMU/ SEI-88-TR-30, September. 1988.

10. An Introduction to Structural Models; presented to the 14th I/ITSEC, November 1992.

11. Structural Models for Real-Time Simulation; training charts from Software Engineering Institute, Carnegie-Mellon University, March 1993.

12. Draft Structural Modeling Guidebook; Software Engineering Institute, Carnegie-Mellon University, and NSIA, April 1993.

ACRONYMS

A/V air vehicle
ACS air combat simulator
ADARTS Ada-Based Design Approach for Real-Time Systems
ADSS Avionics Development Simulation System
ASVP Ada Simulator Validation Program
ATLAS airframe trim linearization and simulation

CDR Critical Design Review
CM configuration management
CMU Carnegie-Mellon University
CNISS Communication, Navigation, Identification Simulation System
CSCI Computer Software Configuration Item

Dem/Val demonstration and validation

E&MD Engineering and Manufacturing Development
EWSS Electronic Warfare Simulation System

FDS flight dynamics simulation
FSL Flight Simulation Laboratory

IPT Integrated Product Team
IRS Interface Requirements Specification
IRSS Inertial Reference System Simulation
ISD Instructional System Development

LASC Lockheed Aeronautical Systems Corporation
LFWC Lockheed Fort Worth Company (formerly General Dynamics, Fort Worth Division)

Nighthawk Harris Computer
NSIA National Security Industrial Association

OFP Operational Flight Program

PCMS Product Configuration Management System
PDR Preliminary Design Review
PTS Pilot Training System
PTSD Pilot Training System Devices

SCR System Change Request
SDD System Design Document
SDL Software Development Lab
SDP Software Development Plan
SEI Software Engineering Institute
SLOC Source Lines of Code
SOF ATS Special Operations Forces Aircrew Training System
SOW Statement of Work
SPE Software Product Evaluation
SPO System Program Office
SRS Software Requirements Specification
S/SEE System/Software Engineering Environment
SSR Software Specification Review
SSS System Segment Specification

TPM Technical Performance Metrics
TS training system
TS SDP Training System Software Development Plan

VDD version description document
VIF VMS Integration Facility
VMS Vehicle Management System
VSS Vehicle System Simulator

WS weapon system
WS SDP Weapon System Software Development Plan

The Heritage of the Air Vehicle Training Systems Domain

David C. Gross and Lynn D. Stuckey, Jr.
Boeing Defense and Space Group

ABSTRACT

One of the Holy Grails of software development has been *reusability*. Everyone is frustrated with continually reinventing the wheel; everyone knows that reuse would dramatically cut costs; and no one has shown an effective reuse paradigm. The trend has been to develop reuse paradigms without regard to past successful projects. Historically, successes with reuse have been accidental -- based on personnel, not on process. Now a new paradigm has emerged that includes a focus on past investments in forming a reuse process. This initiative is DoD's push toward the *megaprogramming* paradigm. Megaprogramming divides system development into two lifecycles, the first focusing on the problem of leveraging assets through a family of related products, and the second focusing on the problem of delivering a single product. The process for the first lifecycle is domain engineering.

Domain engineering is not easy. It revolves around all kinds of questions that simulation software engineers are not used to asking such as: (a) Is this a viable domain?, (b) Is there an acceptably standard partition of the domain?, (c) Is this domain definable?, (d) What granularity is best for domain work products?, and so forth. Yet, if the DoD is going to successfully transition its approach for the development of software intensive systems to the megaprogramming paradigm, software development organizations are going to have to be empowered to meet these challenges.

The U.S. Navy and the Advanced Research Projects Administration are presently funding a megaprogramming demonstration project in the domain of Air Vehicle Training Systems. How has this project come to grips with the technical challenges of domain engineering? Mostly by leveraging the investments of previous research and development projects in this domain such as the Ada Simulation Validation Program (ASVP), the HAVE Module (Mod Sim) Project, the Software Engineering Institute's Structural Model Initiative, the Manned Flight Simulator (MFS), and a series of planned pilot efforts. This paper discusses the advantages and disadvantages on leveraging previous investments into new domain engineering efforts. Its discussion captures valuable lessons about the transition of existing organizational assets into the megaprogramming paradigm.

ABOUT THE AUTHORS

David C. Gross is a software systems engineer with the Missiles & Space Division of the Boeing Defense & Space Group. He has worked in all lifecycle phases of simulation and training systems from requirements development through delivery. He is currently involved in applied research related to megaprogramming in the domain of training simulator systems. Mr. Gross holds a Bachelor of Science in Computer Science/Engineering from Auburn University and a Master of Operations Research at the University of Alabama at Huntsville. His thesis compares the utility of high level languages such as C++ and Ada for simulation. Mr. Gross is a doctoral student at the University of Central Florida, and can be reached at gross@plato.ds.boeing.com

Lynn D. Stuckey, Jr. is a software systems engineer with the Missiles & Space Division of the Boeing Defense & Space Group. He has been responsible for software design, code, test, and integration on several Boeing simulation projects. He is currently involved in research and development activities dealing with software reuse in the domain of air vehicle training systems. Mr. Stuckey holds a Bachelor of Science degree in Electrical Engineering from the University of Alabama, in Huntsville and holds a Master of Systems Engineering from the University of Alabama, in Huntsville. His thesis presents a systems engineering approach to software development. Mr. Stuckey is a doctoral student at the University of Central Florida, and can be reached at stuckey@calif.enzian.com

The Heritage of the Air Vehicle Training Systems Domain

David C. Gross and Lynn D. Stuckey, Jr.
Boeing Defense and Space Group

INTRODUCTION

"The whole of science is nothing more than a refinement of everyday thinking."
Albert Einstein, 1936

Research efforts in software, especially simulation software, have been slow to come to grips with the concept of evolution versus revolution. There are few if any silver bullets in software development, but the industry continues to re-invent the wheel in search of new productivity and quality. These leaps forward are more likely the result of understanding present efforts and then building upon their strengths. If Einstein is correct, then a critical aspect of any project attempting to advance the state of practice must be an understanding of what constitutes "everyday thinking" in that practice. How can we refine something without first understanding it enough to communicate with it? This challenge is further complicated by the diverse nature of technology today -- individual problem domains for completely legitimate reasons adopt approaches which directly conflict with the solutions adopted in other problem domains. Consider for example the Ada language debate -- much of the heat in language discussions arises because the participants fail to acknowledge that different problems have different solutions. Finally, complications arise because the state of the *practice* so badly lags the state of the *art* -- indeed the state of the practice is hardly a tightly constrained range of behavior!

The STARS demonstration of megaprogramming in the Air Vehicle Training Systems (AVTS) domain is certainly trying to advance the state of practice. The STARS approach is a three-pronged attempt to improve the reliability and adaptability of complex software systems. The first prong is *automation* -- the realization that demand for quality software has outstripped our resources to supply it. The second prong is *process* -- the recognition that repeated improvement is only possible when the method of construction is defined and followed. The process in the context of STARS is megaprogramming -- which separates the creation of systems into two distinct lifecycles: domain engineering and application engineering. Domain engineering aims at creating assets for use within a product family; application engineering aims at delivering specific instances of that family. The final prong, and our focus in this paper, is *domain-specific* -- the acknowledgment of the issues raised above.

So in which problem domain are we working? The AVTS domain is a family of air vehicle training devices that provides the simulation, stimulation, and/or emulation of all the components and systems for a real-time air vehicle simulation. [STUCKEY] This domain encompasses the systems necessary to provide training devices that a trainee uses to become familiar with the configuration and/or flight characteristics of an application air vehicle, gain proficiency in executing normal procedures, recognizing malfunctions/abnormal indications and executing the corresponding standard/emergency procedures, and executing mission procedures. The devices, or *instances*, of this domain are some proper subset of the domain. This domain includes the system diagnostic and test requirements for the applicable air vehicle devices based on individual segment requirements.

But this work is not occurring in a vacuum, as illustrated in Figure 1. Indeed, where it not for the rich

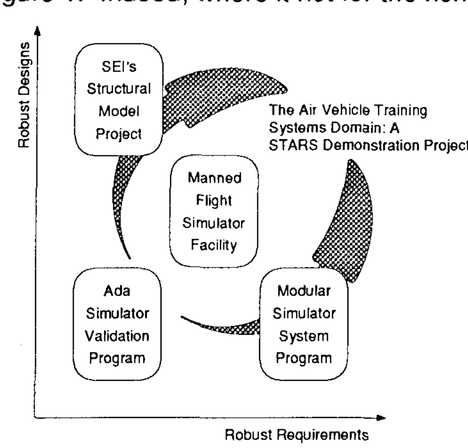


Figure 1: Projects Leveraged in the AVTS Domain

resources developed by earlier research in the field, this project could not be attempted -- the problem domain would simply not be sufficiently mature to make the attempt with constrained resources. This paper discusses some of the research in the domain that makes this project possible, and the challenges in incorporating their results

HERITAGE PROGRAMS

Ada Simulation Validation Program (ASVP)

Synopsis.

The ASVP was an Air Force contracted research and development program. The program spanned some 24 months from 1985 to 1987 with extensions that followed. It was part of the continued effort by the DoD to have contractors demonstrate the validity and application of the Ada programming language in a variety of environments.

The task for this project was to re-develop in Ada, a substantial portion of the application software for the existing E-3A Flight Crew Trainer (FCT), and to answer the following questions: Does Ada work on simulators? Is Ada better or worse than FORTRAN for simulators? What are the better methods for designing simulator software in Ada? What software development and support tools are necessary in the development lifecycle?

ASVP was a 24 month program consisting of some 23,000 man hours. Boeing was teamed with SAIC and Encore. The E-3A simulator re-developed was fielded at Tinker Air Force Base. The software was re-developed and tested in Huntsville Alabama, while hardware/software integration, system test, and demonstration occurred at Tinker. The work at Tinker was accomplished on second and third shifts. This schedule permitted E-3A crews to continue training with no disruptions to their training mission.[ASVP-FR]

Contribution.

ASVP was an overwhelming success. It received the first YW Systems Quality Award. The system passed 87% of the acceptance test procedures on the first pass. The FCT was evaluated by Air Force pilots with over 4000 flying hours in an E-3A and by instructors who trained pilots daily. The program also became the basis for future Air Force simulator work. It provided guidelines for the implementation of structural models, coding standards, and an object-oriented design methodology for simulators.

Modular Simulator System Program (Mod Sim)

Synopsis.

The Mod Sim program was a tri-service supported development program. The primary goals of the modular simulator design were to shorten simulator development schedules, reduce simulator development costs and improve simulator supportability. The program was organized into three phases. Phase I surveyed the industry as to the desirability and feasibility of introducing a generic modular simulator concept. Phase II, Modular Simulator Design Concept Development, produced a conceptual modular simulator architecture with a focus on aircrew simulator functional analysis and intermodule communication architecture/design. The contractors developed a conceptual modular simulator design for this effort. Phase III consisted of design, demonstration and validation of the modular simulator concept. To foster industry participation and "buy in" to the Mod Sim design, Boeing was required to subcontract the design and development of 50 to 75 percent of the segments. The Phase III subcontractors were Rediffusion Simulation Limited (RSL), Science Applications International Corporation (SAIC), AAI, and Intermetrics. To gain further industry participation, regular Interface Standards Working Group (ISWG) meetings were held. At these meetings both industry and government simulation experts were allowed to participate in the review of the modular simulator design and subsequent demonstration.

Phase III was divided into two parts. Part 1 accomplished four major tasks:

a. System Partitioning. This task involved the analysis of simulations for a large number of fixed and rotary wing training devices. This data, along with other raw data and the conceptual partitioning from Phase II were used to create a Functional Dictionary that contained an allocation of all functions and the interface requirements between functions. The Functional Dictionary and segment partitioning were refined through an iterative process using an Artificial Intelligence tool. This resulted in segments that had generic intersegment interfaces, were loosely coupled, and focused on a specific area of simulation expertise.

b. Communication Architecture. This task involved the specification of a hardware and software communication architecture that would allow the segments to communicate effectively.

c. System Performance Model. In order to efficiently select a communication architecture a System Performance Model was constructed to emulate the various design alternatives. Fourteen data buses and seven protocols were analyzed.

d. Specifications. To promote the standardization of the Mod Sim architecture, a thirteen volume generic System/Segment Specification (DOD-STD-2167A) was prepared. The system level specification defines the communication architecture and requirements common to all segments. The segment level specifications define the unique requirements applicable to the segment.

During Part 2 of Phase III, Boeing and the subcontractors demonstrated the Part 1 design. The demonstration was accomplished using a government provided F-16 crew station and existing F-16 simulator source code. The government furnished products were adapted to the modular simulator partitioning and communicated using the modular simulator communication architecture.

At the completion of the Phase III, several follow-on tasks were contracted. These consisted of adding an interface to the Mod Sim architecture to support multiple simulator/team training (e.g.; Distributed Interactive Simulation), adding tailoring instructions to the generic specifications to ease adaptation to specific applications, and the creation of Mod Sim guidance documentation. This documentation included an engineering design guide, a management guide and an executive report that provides an overview of the Mod Sim approach. The Mod Sim architecture is shown in Figure 2. The architecture consists of 12 distinct segments that communicate via a Virtual Network (VNET). [MSS-MGT]

Contribution.

There are several distinct advantages to using the modular simulator design and design concepts in developing training devices. These advantages include:

a. Systems Engineering. The Mod Sim design provides a wealth of generic systems engineering products that are reusable for any application. This reduces front end development cost and schedule and mitigates risk throughout the project.

b. Subcontracting. One of the primary requirements for the Mod Sim architecture was the capability to independently specify, develop, and test individual segments as stand-alone products. This enhances the ability to subcontract development of segments by providing well-defined interfaces that reflect a straight-forward allocation of simulator functions along traditional subsystem

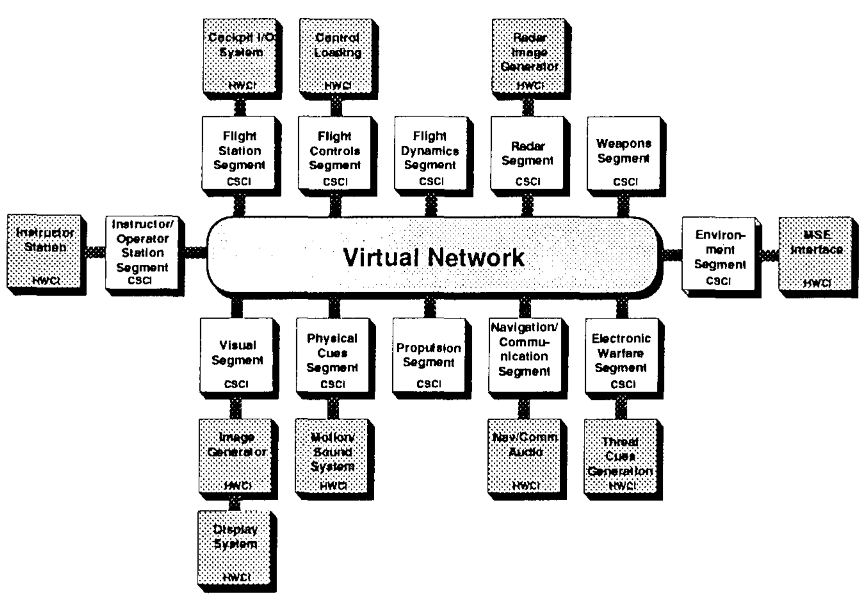


Figure 2: Mod Sim Architecture

product boundaries, to make best of advantage of individual organization's strengths.

c. Integration. Use of the Mod Sim architecture and strong Ada design principles significantly reduces integration time. This has been proven in a series of demonstration projects.

d. Reusability. The Mod Sim architecture promotes and enables reuse among families of training devices and applications. Experience has shown that architecture is the key to successful higher order reuse.

e. Design Flexibility. The Mod Sim architecture allows latitude in design to support low cost and high cost devices. The Mod Sim architecture does not place any requirements on the internal design of the segments.

f. Parallel Development and Stand-alone Testing. Mod Sim segments can be developed and tested in parallel due to the well defined segment requirements and intersegment interfaces. This can significantly shorten the overall system development schedule and reduces integration risk by eliminating common interface problems early in the development and testing phases.

SEI's Structural Model

Synopsis.

The contract work to define a structural model and the concept of structural modeling has been a collaborative effort between the United States Air Force, it contractors, and the SEI. A structural model is an application framework for flight simulators and the structural modeling process is the means by which this framework is engineered into a complete system. The recognition of the technical risks associated with building complex systems such as flight simulators was the catalyst that drove the development of the structural modeling method. The broad objective behind structural modeling was to take a complex problem domain and abstract it to a coarse enough level to make it manageable, modifiable, and able to be communicated to a diverse user and developer community. Structural modeling experience has been gained in a number of recent simulator acquisitions, including the B-2 Weapon Systems Trainer, the C-17 Aircrew Training System, and the Special Operations Forces Aircrew Training System. In addition, the Real-Time Simulators Project at the SEI is currently drafting a guidebook describing in detail structural modeling as it applies to the development of an air vehicle within a flight simulator, specifically addressing the case study of the T-39A flight simulator. Figure 3 illustrates the Air Vehicle Structural Model. [ABOWD]

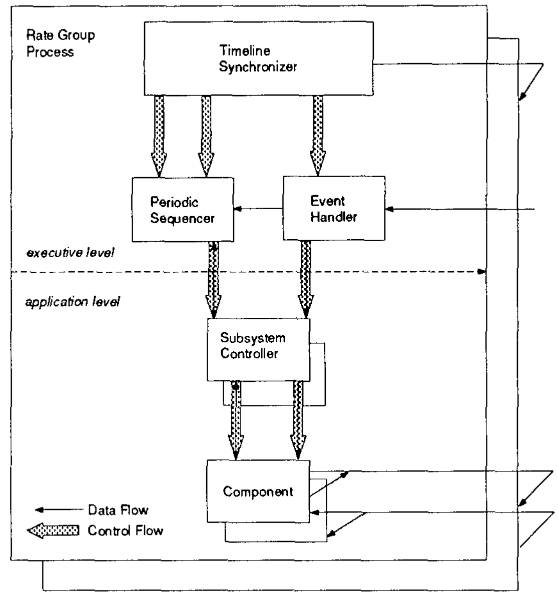


Figure 3: SEI's Air Vehicle Structural Model

Contribution.

Specific benefits realized in flight simulator software development from the structural model described are:

a. Increased separation of the coordination model from partitioning strategy,

b. Easier integration of independently developed software components, and

c. Stronger classification of systemic and (particular) mission requirements.

The Structural Model project determined that considerable design reuse is feasible; whereas code reuse is restricted to the level of the component. The project concluded that modeling mission requirements is more volatile than modeling the other requirements, and it is appropriate that they are no longer the main drivers of the software design.

Manned Flight Simulator (MFS)

Synopsis.

The U.S. Navy, in response to every increasingly risky and costly flight test of new aircraft, decided to create Manned Flight Simulator(MFS). The goal of MFS was to provide to the Navy a low cost high fidelity simulation capability which would allow for growth. The Navy created a laboratory facility and funded the development of a highly modular engineering software architecture. The architecture was created and called Controls and Simulation Test Loop Environment(CASTLE). CASTLE is a highly modular simulation system which affords the simulation engineer the ability to create models with the knowledge that his equations of motions, environment, hardware and visual environment interface are all predefined. The pre-definition and reusable modules allow the simulation engineer to concentrate on the functionalities of his math model. The manned flight simulator has a long history of opportunistic reuse of such airframe models as, F-14, F/A-18A, T-45A, AV-8B, AH-1W, V-22, UH-60. All of these airframe models were obtained from outside sources and rehosted to the CASTLE system. Upon rehost, the NAVY was able to apply advanced engineering and analysis tools to increase the fidelity of these models. This opportunistic reuse has saved the Navy money and created a simulator environment that works in close conjunction with the flight test personnel. [PRYOR]

Contribution.

MFS has evolved techniques for developing simulation models that are abstracted from the underlying hardware. It is clear that many simulation models are completely independent of hardware, for example, the atmosphere model. However, it is more difficult to isolate other models to a minimum of dependencies. MFS has established that this paradigm shift can be made, and real benefits for reuse flow from it.

Other Relevant Projects

There are several other DoD initiatives that involve the standardization of simulators and training systems. Examples include Distributed Interactive Simulation, Project 2851 Database Standardization, the Simulator Data Integrity Program, and the Universal Threat System for Simulators. AVTS does not constrain the use of these standards. In fact, some characteristics of the architecture were designed to accommodate or enhance compatibility with these standards, in so far as publicly available materials permit.

LEVERAGING LEGACIES INTO DOMAIN ENGINEERING

It is important to understand our perspective on why the use of legacy assets is important. To this end, a description of our process, level of reuse, and utilization of legacy assets is appropriate.

Megaprogramming

Over the years, the STARS project has focused on enabling a paradigm shift of DoD software practices to megaprogramming. The central megaprogramming concept is a process-driven, two lifecycle approach to software development. One lifecycle spans the creation and enrichment of a family of related product, or *domain* (see Figure 4). The other lifecycle spans the construction and delivery of individual products to customers, or instances of the domain. The STARS project uses the Virginia Center of Excellence's (VCOE) process called Synthesis for detailing the separate lifecycles. Synthesis defines the effort associated with creating the assets as *domain* engineering, and the effort in creating a specific product as *application* engineering. With a legacy asset perspective, domain engineering is the important lifecycle.

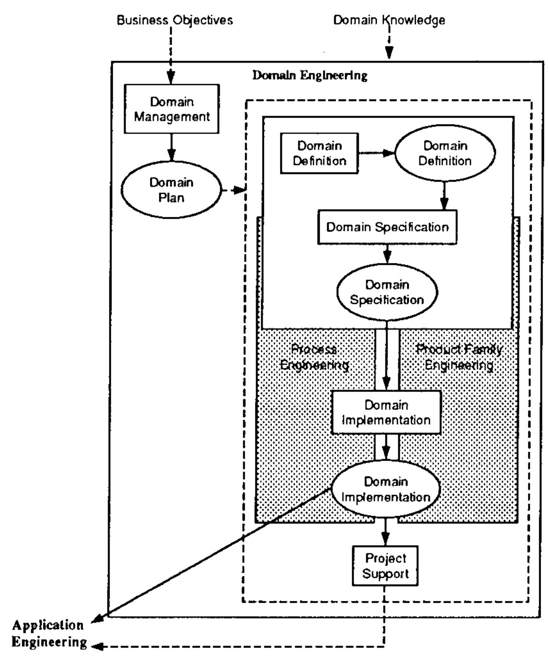


Figure 4: Domain Engineering

Level of Reuse

Synthesis is a process developed for *leveraged* reuse. Leveraged reuse is one of five approaches to reusing software: (a) ad hoc, (b) opportunistic, (c) integrated, (d) leveraged, and (e) anticipated. Leveraged reuse assumes that a given product is actually a member of a product family, the members of which share some degree of commonality. Synthesis involves the definition, analysis, specification, and implementation of a domain which encompasses a *viable* product family -- a family which shares sufficient commonality (and predictable variability) to justify an investment in the domain. Individual products are developed as instances of the domain, which reuse common elements of the domain and adapt variable elements using a defined, repeatable process.

Given our focus on leveraged reuse, legacy assets become increasingly important. They are the basis for the structure, evolution, and *product fragments* held within the domain. Without prior research in the field, the AVTS domain would be non-viable.

Application of Legacy Products

Figure 5 describes the usefulness of legacy assets within the domain engineering cycle of the Synthesis process. It is evident that these assets have a major role in most activities. In domain management they form the basis for viability assessment, evolution planning, and risk analysis. In domain definition the assets are used to bound the domain, validate terms, and define assumptions. In domain specification they are used in requirements analysis and product design. In domain verification they provide historical context to evaluate correctness. In product implementation the assets are used for algorithmic development as well as entire module utilization. In domain validation they are used as a measure of effectiveness. Finally, the assets are utilized as an historical perspective on domain delivery.

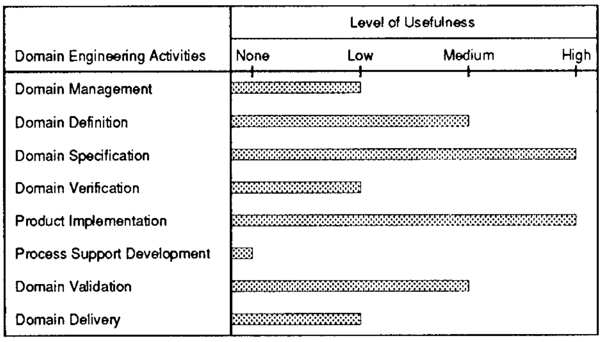


Figure 5: Usefulness of Legacy Assets

LEVERAGING PROGRAMS

The attempt to incorporate new technologies, such as megaprogramming, into the state of the practice is fraught with peril. Figure 6 illustrates the process (which by itself is a example of building upon prior research -- this is a chart from ASVP with Megaprogramming substituted in for Ada). The existence of useful prior research presents a paradox: prior research created the critical mass to make the attempt, but using such research can open a Pandora's box of problems. The following discussion outlines the pitfalls and heuristics for such use.

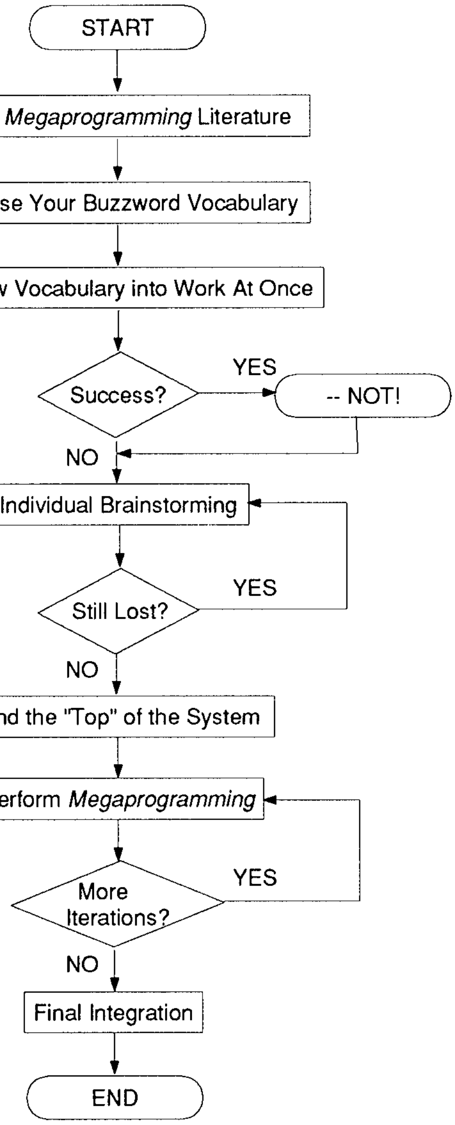


Figure 6: Adopting Megaprogramming

Pitfalls

Our experience has uncovered some of the pitfalls of building upon research legacies.

❑ Making the Leap of Faith

Every demonstration project is organized around some central "great concept." The great concept for the STARS demonstration project in the AVTS Domain is megaprogramming For the purposes of the project itself, it is (surprisingly) not so important whether or not it *actually is* great -- the purpose of the demonstration project is to evaluate that very premise. In fact, the participants on the project must be able to maintain a *suspension of disbelief* about the worthiness of the concept in order to make their very best effort at making it work. This is analogous to the difference between "pure" science and "pure" engineering. Failed science experiments are not surprising and even expected -- but failed engineering projects are failures for which (generally) someone is held accountable. Science (and engineering research) is a venture into the *unknown* -- engineering is the application of *known* mathematical and scientific principles to practical ends. Therefore, the productive participant must maintain this mindset -- but this attitude alone is not enough! The productive participant must also be prepare to re-evaluate the most fundamental principles in light of the great concept. He/she must be able to conceive and implement a paradigm shift -- discarding long held and favored practices in so led by the great concept. This requirement of suspending disbelief and questioning principles is a trap which captures many fine engineers. Time, and time again, we have seen perfectly good engineers who were unable to make this leap of faith.

❑ Quoting Scripture

The great concept generally arrives in a holy book developed in some other context. Sometimes, it arrives in a well-defined specification e.g., MIL-STD-1815A in the case of ASVP). In the case of the STARS Demonstration in the AVTS Domain, the great concept was defined by the Reuse-Driven Software Processes, developed by the VCOE. The guidebook (along with companion material and help from VCOE) has proved extraordinarily useful in proceeding with the demonstration project. The pitfall is that while the guidebook is a useful first approximation , project participants are tempted to argue from it as revealed truth. Rather than arguing the merits of a particular position, we frequently found ourselves saying "but, the guidebook says..." Relevant and understandable examples prove much more powerful than quoting the guidebook -- its use is in provoking thought.

❑ Ivy Tower Meets the Gridiron

Just because the concept is expressed does not mean it is fully fleshed out. Often, the painful details are "left as an exercise for the student". Organizations frequently respond to such situations with lip service -- how many times has empowerment become a euphemism for no responsibility. How many projects have claimed adherence to some hot technology (e.g., object-oriented) not because they understood it and believed in it, but because it sounded so good.

❑ Not Invented Here Syndrome

If we are to make use of any legacy projects, there must be kernel of truth in them that project participants are able to utilize. Of course, just because a research project has occurred does not mean it is prima facia useful -- but the more frequent difficulty is that some people have Not-Invented-Here syndrome. While skepticism is a natural and healthy attitude for an engineer to bring to the latest "revolutionary solution", refusal to consider the possibility of some advance is the death knell for a research effort. If the project can not persuade participants to overcome this attitude, then the legacy *can not be used* no matter how significant their potential application.

❑ Leveled Experience Base

One dilemma that arises in a research project is that the various participants come with the baggage of their own experiences. Each individual sees this project in terms of the ones he or she has worked most often. The AVTS domain includes some natural examples: consider the level of complexity differences between a part-task instrument trainer and a full capability weapon system trainer. Some of our project's participant expect a software system size on the order of 15,000 lines of code -- and at the other extreme are those that expect any "real" trainer must have at least a 1,000,000 lines of code. Obviously, your expectations about size impact your approach to the problems of developing software in the domain. Once again, just as specific trainers are different, so are specific individuals -- the project must be able use and apply this diverse experience base.

❑ The Vision Thing

The empowering premise of a demonstration project is the promise it brings. For example, the Mod Sim project held out the promise of avoiding repeating high cost analysis, and building in the expectation of a diverse contractor base. However, the glitter of the promise is different in every participant's eye. The project *must* be able to create and communicate a common vision for the future. Failure to do so has two major disastrous outcomes. First, while participants are still ignorant that their version of the vision is not the "right" version, they will waste resources working hard in conflicting directions. Second, things get worse when they find out their vision is not the "right" version. Most people will feel a sense of betrayal. This can happen at any level -- engineers or management or project sponsors.

Heuristics

Our experience supports the assertion of a set of heuristics from building upon research legacies. Application of these heuristics will conserve the scarce resources of a research project.

❑ Accept Overlapping Legacies

Our first heuristic is that when various legacies overlap in the approach to some aspect of the domain, the project can productively accept the overlap as "proven" in the context of the domain. The easiest example is Ada -- three of this domain's four research legacies adopted Ada, so the AVTS domain accepts Ada as its language of choice. Few project resources were invested in consideration of the language choice issue.

❑ Seek Synergies Between Legacies

Next, we suggest that when the research legacies seem to complement each other, there are synergies that the project can take advantage of while minimizing the required resource investment. For example, all of the AVTS research legacies pointed at a controlled, hierarchial, architecture with controlled communication -- hence AVTS's adoption of the Domain Architecture for Reuse in Training Systems (DARTS). DARTS is *not* the architecture of any of the research legacies but derives from the best aspects of all of them.

❑ Architecture as the Glue

A heuristic that derives from our participation on all of these research projects is that architecture is the glue that holds the project together -- at least for domains involving complex software systems. If the project can achieve consensus on an architectural approach which reflects the needs of that particular project, the architecture serves as the critical context within which participants interpret research developments. On a aggressive project, it is easy for participants to be overwhelmed and left out of the loop -- architecture is an important part of avoiding this disaster.

❑ Controlling Concept Introduction

This heuristic relates to self-constraint -- it is not so important that a given project adopt every idea in the marketplace as it is that the project understand them in its own context. Any research project wants to be at the cutting edge in every aspect of its domain -- if for nothing else to protect its own credibility. But the volume of potentially useful ideas is so great that a project can easily gorge upon the feast and choke to death. Visionaries on the project tend to seek to integrate everything, without regard to how tenuous its relationship to this project. Since the resources of demonstration projects are even more limited than for delivery orders in this age of declining budgets, a project can not be successful by trying to do everything. Everyone must "play by the rules", that is, focus on the objectives of *this* project. Healthy projects will carefully observe the rapid river of research, and *decide* what to fish out rather than swallowing it all.

❑ "Prophets" Vs. "Priests"

This heuristic points out that while the current project is only possible in the light of what has gone before, the current project is fundamentally *different*. The participants on a research project have a mini-lifecycle all their own. When a project is young, the participants are like prophets, challenging the existing order and predicting the future. But as a project matures, its participants become more like priests, protecting and defending the faith against the barbarians at the gate. This is particularly dangerous when a project attempts to leverage a legacy by bringing participants of earlier projects into the new research. Such individuals may have trouble adjusting their mindset to the new challenge. Consider for example one difference between the Mod Sim challenge and the AVTS challenge. Mod Sim's requirement was to build tailorable products; AVTS's is to build adaptable products (and processes). This seemingly trivial shift of nomenclature has extraordinary impact throughout the project. Consider for example the difference between defining an interface between components that can be tailored (i.e., rewritten) by some end user, and an interface that must automatically adapt to an end-user's specific requirements. It is not hard to imagine someone "stuck" in the Mod Sim mindset and unable to correctly address the AVTS challenge.

CONCLUSION

> *"If I have seen further it is by standing on the shoulders of Giants."*
>
> Issac Newton, 1675

Newton was referring to the tremendous advances made by mathematicians, physicists, and astronomers such as Da Vinci, Copernicus, Galileo, and Kepler. At its best, science is the most constructive of human endeavors, with each development building carefully upon the discoveries and inventions of earlier generations. But two things have increased the difficulty of scientific advances in our times. First, no longer do we build upon the shoulders of individual giants, we must build upon the accomplishments of teams of giants. While this is a direct outcome of the complexity of the challenges we are addressing, the product of teams are typically much more amorphous and difficult to quantify -- hence the aphorism, "a camel is a horse designed by a committee." When confusion about the use of an individual's product arises, one consults the individual for a (generally) consistent explanation. But teams are notorious for lack of shared visions, so explanations from different team members frequently conflict.

The second difficulty arises from the size of the technical marketplace. When Newton developed his calculus, he faced competition from perhaps two serious contenders, in a scientific community of certainly less than 10,000 members. In contrast, Capers Jones reports that there are some 1,818,500 programmers working in the United States alone, each one with no doubt strongly held opinions about the "right" language, methodology, tools, etc. Two orders of magnitude more people working on a problem that must be three orders of magnitude less pervasive. Consider the debates in our community about the suitability of Ada -- but Jones says that the leading language in use today is still *COBOL*. Ada fails to make the top five and is lumped in with "all other languages"! How can any purported advance make headway in this tower of Babel? [JONES]

We can advance the state of the practice, not by re-inventing the wheel, but by carefully building upon work which has gone before. Software has been called by many others the most complex endeavor attempted by mankind -- the likelihood of a single individual uniquely making a significant discovery is negligible. The complexity of the problem domain demands teams of world class experts building upon the work of other such teams.

REFERENCES

[ABOWD] Abowd, Gregory D., et. al., *Structural Modeling: An Application Framework and Development Process for Flight Simulators*, CMU/SEI-93-TR-14. August 1993. Software Engineering Institute, Pittsburgh, PA.

[ASVP-FR] *Ada Simulation Validation Program Final Report.* 9 September 1988. The Boeing Company, Huntsville, AL.

[JONES] Jones, Capers. *Gaps in the Object-Oriented Paradigm*, IEEE Computer. June 1994. IEEE Computer Society.

[MSS-MGT] *Modular Simulator System Management Guide*, D495-10439-1. 13 September 1993. The Boeing Company, Huntsville, AL.

[PRYOR] Pryor, Greg. *Personal Communications.* June 1994. Former lead aerodynamics engineer of the MFS AH-1W Aircrew Procedures Trainer.

[STUCKEY] Stuckey, Lynn, et. al., *Technical Expectations of a Full Scale Domain Engineering Demonstration Project.* November 1994. The 16th Interservice/Industry Training Systems and Education Conference.

CUSTOMIZING AN OBJECT-ORIENTED DESIGN OF LEADSHIP EFFECTS

Jerome M. Weiss
Systems Engineer
CAE-Link Corporation

ABSTRACT

An Object Oriented Design (OOD) approach to the simulation of Leadship effects was presented at the 1991 IITSEC Conference. These Leadship effects were coded in Ada, contained within the Othership Subsystem and used on the B-2 Aircrew Training Device (ATD). This paper will illustrate the success of this subsystem's structure during subjective pilot evaluations and limited flight test data correlation. These Leadship effects were necessary to provide realistic aerial refueling and base escape training. The Othership subsystem was structured to be generic in form, highly transportable and easily maintainable.

Subjective pilot evaluations and a limited flight test data correlation have been performed for the training task of aerial refueling. The ease of conducting these evaluations and correlation support the 1991 stated advantages to this subsystem's structure. Ease of maintainability was demonstrated by customizing this subsystem for two different tankers (KC-135R and KC-10A) with tanker-unique data modifications within the same evaluation session. High transportability or reusability was shown by customizing only the significant leadship effects and eliminating the insignificant leadship effects without altering the subsystem's basic form. Generic engineering notation supported both maintainability and reusability. The short time required to customize this subsystem is additional support of the lessons leaned. Additional lessons learned during these evaluations and correlation has led to a second generation OOD structure for the Othership subsystem.

The second generation OOD structure would be the perferred architecture of an Othership subsystem slated for a training device with similar training requirements.

About the Author

Mr. Weiss is currently a lead engineer for the Air Vehicle department on the B-2 Aircrew Training Device. Previously Mr. Weiss led the Aerodynamic Simulation for the C-135B and B-52H trainers. Prior to joining CAE-Link in 1980, Mr. Weiss was employed with Cessna Aircraft Co., Boeing Military Aircraft Co. and the Air Force Flight Test Center. Mr. Weiss holds a M.S. in Mechanical Engineering from CSU at Fresno (1978) and a B.S. in Aero-Space Engineering from SUNY Buffalo (1973).

CUSTOMIZING AN OBJECT-ORIENTED DESIGN OF LEADSHIP EFFECTS

Jerome M. Weiss
Systems Engineer
CAE-Link Corporation

INTRODUCTION

Realistic training of maneuvers that include an air mass disturbed by a preceding vehicle (leadship) was a requirement for the B-2 Aircrew Training Device (ATD). The training of Aerial Refueling and Base Escape (Minimum Interval Takeoff) were part of the original B-2 ATD Prime Item Development Specification (PIDS). This was satisfied by the functionality of the Othership subsystem. The Othership subsystem utilized the techniques of Object Oriented Design (OOD) which emphasized maintainability and reuseability. The original subsystem was initially presented in ref. 1.

This paper presents an overview of the original simulation problem and its solution. World events and technical considerations led to changes to the original subsystem. These changes manifested themselves into two categories which were addressed during the customizing of the Othership subsystem.

Customizing the original Othership subsystem addressed most of these changes. The initial customizing process was performed without altering the basic format of the original OOD subsystem structure, and within the original partitioning scheme. Several features of the original subsystem structure that were intended to improve the customizing process lived up to expectations.

Addressing the remaining area of change required an investigation of an advanced partitioning scheme. This investigation led to a second generation OOD structure for this subsystem. The second generation Othership subsystem is capable of addressing all areas of change.

This paper will also present the lessons learned during the customizing process, as well as conclusions and recommendations.

Original Othership Subsystem

Aerial Refueling and Base Escape maneuvers were required for the B-2 ATD. Both of these maneuvers require the B-2 to fly behind another aircraft. During these maneuvers the aerodynamic interference of a "preceding" aircraft (leadship) can be felt by the "following" aircraft (lagship). This was the original simulation problem. In the B-2 ATD the aerodynamic interference originally consisted of the following functionality: leadship wingtip vortex and downwash, leadship engine exhaust and lagship bow wave drag. All three of these effects would vary with different leadships. The B-2 ATD PIDS mandated that the leadship be either a tanker (KC-135R, KC-10A) or another B-2.

Compliance with the above training requirements led to the original simulation solution (Othership subsystem). The structure of the Othership subsystem was created using Object-Oriented Design (OOD) techniques and allowances for the limited availability of application specific design data. It also used a partitioning scheme that was already in place. This partitioning scheme separated the B-2 ATD into various functional subsystems. These subsystems were defined historically by functional decomposition, and in some areas the format of the design data (e.g. Weight and Balance Subsystem, Aerodynamic Coefficient Subsystem, Equations of Motion Subsystem, Navigation Subsystem, etc.). The Othership subsystem was created after it was decided that all other flying platforms (e.g. tankers, friendly/foe aircraft, etc.) are housed in a separate subsystem called "Targets" subsystem. The fact that the Othership subsystem was defined after most of the partitioning of subsystems occurred had a influence on its contents.

The subsystem was made up of self-contained objects that reflected logical/physical real-world entities. The functionality of these objects were represented in ADA code within separately compilable units and their subunits. Three objects were defined to make up this subsystem: Leadship, Air and Lagship. Reference 1 contains more details pertaining to these objects and their attributes.

The original and current Ada package structure is shown in Figure 1 which includes the data Imports package and local data Declarations package. The Imports package contained the externally required data from other subsystems. The Declarations package serves as a central point for initialization for all parameters and constants.

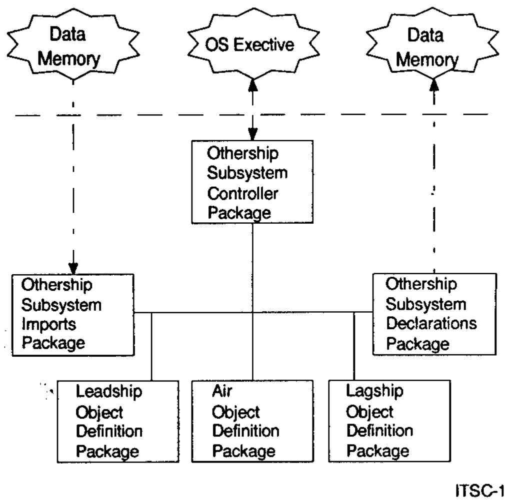


Figure 1 Original Othership Subsystem Ada Package Structure

The subsystem controller was designed to select which models (e.g. engine exhaust) were needed to be executed during various simulation states, such as Real-Time, Reset, Freeze, or Initialization. Within certain simulation states logic was added to review critical parameters, such as relative position, to assure that only the necessary code was executed. This logic is referred to as model and/or model component control logic. It resulted in the minimization of the execution time of this subsystem.

All of the object's equations were coded using generic engineering notation and adjustment constants, instead of application-specific notation. An example of these two notations is shown below:

Generic Engineering Notation	Application-Specific Engineering Notation	Definition of Constants and Geometry
$D=(qSC_D)K_1$	$D=1200qC_D$	$S=2400\,ft^2$, $K_1=0.50$
$y=(\frac{\pi}{4})b$	$y=74.6$	$b=95\,ft$

More details about the Othership subsystem object selection, structure, object attributes and calculations can be found in ref. 1 and 2.

External interdependencies for aerodynamic interference calculations come mainly from two other subsystems: Visual Interface and Targets. The results of the aerodynamic interference calculations are exported to the B-2 ATD Forces and Moments subsystem. Figure 2 portrays the major Othership Subsystem's external interfaces.

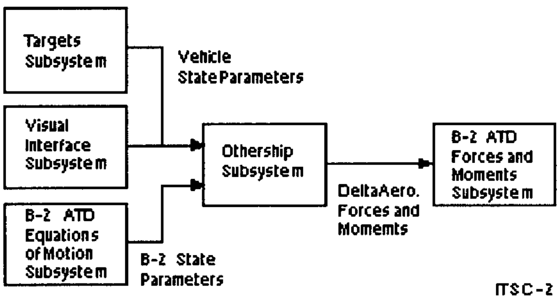


Figure 2 Original Othership Major External Interfaces

Why Customize the Original Subsystem

The original simulation subsystem contained a solution to the original mid-1980s simulation problem based primarily on generic design data. Since then three events occurred that caused changes to the original subsystem. These events were:

1. Changes to the training requirements - Removal of the Base Escape training requirement. This resulted from changes in the world's political climate.

2. System level integrated testing inputs - High level testing of the B-2 ATD's performance during aerial refueling led to three troublesome areas:
   a. Leadship's response to winds
   b. B-2 ATD's throttle response characteristics.
   c. Certain aerodynamic interference functionality was unrealistic and not necessary for this training.

   This testing included a limited flight test correlation and inputs from non-B-2 crew members with experience at aerial refueling.

3. Subjective evaluation inputs - B-2 crew members evaluated aerial refueling in the B-2 ATD. Similar issues to the ones found in the system level integration testing were found here too.

The events listed above produced four changes to the original subsystem. The four changes were categorized into two groups:

1. Anticipated Changes - Aerodynamic interference performance issues related to the characteristics of certain functionality.

   Note: Certain aerodynamic interference performance changes were expected because of the limited B-2 specific design data in this area.

2. Unforeseen Changes:
   a. Removal of the Base Escape training requirement.
   b. Aerodynamic interference issue related to the removal of the leadship exhaust effects because of its low training value.
   c. Leadship (Tanker) response to various types of winds (gusts, cross winds, etc.) was uncharacteristic of the real world. This was the results of selecting overly simplified equations of motion selected for the leadship.
   d. B-2 ATD response to throttle movements did not accurately represent that of an actual B-2. This is due to limited throttle transient validation data. This was also found to occur in other maneuvers such as Landing. Therefore, it was handled within the Propulsion subsystem.

CUSTOMIZING THE ORIGINAL SUBSYSTEM

This paragraph is broken down into three sections. The first section will describe the limited success of customizing the original subsystem with the original partitioning scheme. The second section contains testing benefits realized from the original subsystem. The last section describes a second generation OOD structure for the Othership subsystem. It utilizes an advanced partitioning scheme that addresses all Othership changes.

Customizing With the Original Partitioning

Attempts to resolve both categories of changes were made by customizing the original OOD simulation solution under the original partitioning scheme. Three of the four changes were successfully solved by this customizing effort. The subsystem controller's model control logic was modified to account for the changes related to removing functionality. Since removing the functionality of Base Escape and Leadship Exhaust effects is application specific, the logic modification was made to support reuseability. The underlined logic of Figure 3 "and Required", illustrates how the logic was modified. Figure 3 also shows that this was accomplished without changing the format of the original subsystem's structure.

Modifying the characteristics of certain functionality took advantage of generic engineering notation. The modifications consisted of reinitializing adjustment constants, modifying the magnitude of existing functions and introducing additional functions to existing generic equations. Modifications to the software were made at the lowest appropriate level. Figure 3 illustrates the number of units that exist within each model and the number of those units requiring modification (e.g. 1 of 10). As an example, consider the modification to the aerial refueling model. Updates consisted of modifying an existing performance function and reinitializing some adjustment constants. This allowed the basic equations to remain unchanged and totally generic, yet they yielded improved performance for each different type of leadship. Only one of this model's ten compilation units required modification.

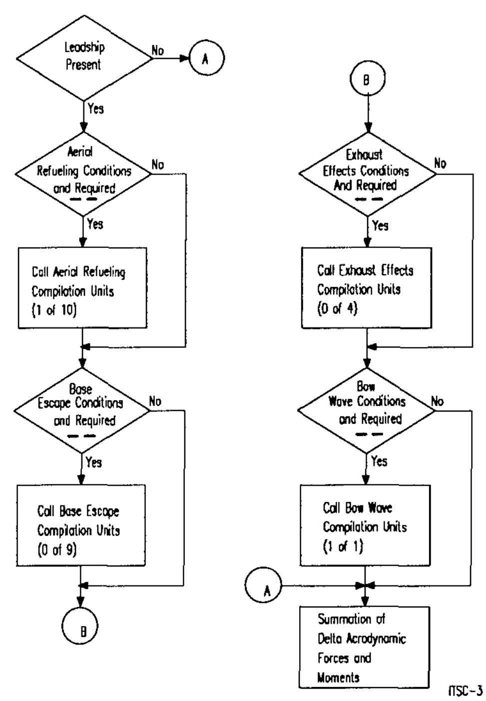


Figure 3 Customized Othership with Original Partitioning

Benefits During Testing

The structure of the original OOD subsystem took into account the predicted need for responding to high level testing and subjective inputs. The ability to attack performance problems at the lowest level (lowest compilation unit) also prevented the accidental corruption of software units unaffected by the customizing effort. Regression testing was minimized because the original generic equations did not change. Only the modified functions and adjustment constants needed to be accounted for. It took only a couple of hours to code and regression test a software change slated for incorporation into the trainer. This meant that the changes were ready in time for the trainer's next daily software update.

The generic equations allowed for simplified testing tools because parameters of interest were independent of a particular application. The same tools would work regardless of the leadship being utilized.

Three aerial refueling subjective evaluations were performed. No single session exceeded 3.5 hours, and covered two different leadships (e.g., KC-135 and KC-10) at various flight conditions. Within these sessions improved aerial refueling characteristics were accomplished. This compared favorably to a previous trainer, which required twice as many hours to accomplish the same results.

Customizing With the Advanced Partitioning

One unforeseen change still remained unsolved after completion of the customizing with the original partitioning. The original partitioning scheme did not facilitate the most reusable solution to the uncharacteristic leadship response to various winds. A more advanced partitioning scheme needed to be investigated. This investigation has led to a second generation OOD simulation solution. The new Othership subsystem now includes the kinetics of the leadship, as well as the previous leadship functionality discussed in ref. 1. This gives the Othership subsystem greater control over the training maneuver. Greater control will allow for quicker response to high level testing results and subjective inputs. This also translates into shorter testing time and increased maintainability.

The advanced partitioning scheme takes advantage of a software reuse library. This library contains existing functionality, already designed, coded and tested in a generic format. The functionalities are identified as "Classes" and also will usually contain "subclasses". The definition of a "Class" used within this paper is "a group of objects with the same state data structure and same behavior". Examples of a "Class" may be a Gas Turbine, Motor, Electrical Control Circuit, or the Vehicle Kinetics. Associated "subclasses" of a Gas Turbine could include the Compressor or Fuel Controller, and associated "subclasses" for Vehicle Kinetics could be Accelerations, Velocities, or Positions, It should be stated that "Classes" are required to be broad in scope. Therefore a "Class" may contain functionality at various levels of fidelity. It then becomes the job of a specific application or instance of a "Class" to select the required fidelity. Reference 3 contains more detailed information related to this advanced partitioning scheme.

It is important to realize that any "Class" may be used by multiple subsystems. Also realize that "Classes" and/or "subclasses" could be reused multiple times within a subsystem. Figure 4 shows an example of this concept. The same "subclasses" of Position and Velocity can be called by different subsystems (Targets, Ownship Kinetics and Othership), and have multiple calls within a single subsystem. The required fidelity will be determined internally within each subsystem. Another advantage of using "Classes" is that it shortens the design, code and test time in the assembly of a trainer.

The second generation Othership subsystem structure is presented in Figure 5 utilized four "Classes": Air, Vehicle Kinetics, Leadship Aerodynamic Interference and Lagship Aerodynamic Interference.

A brief description of their functionality is presented below:

> Air - Contains all atmospheric functionality. The Othership subsystem would use only a small part of this total "Class". It would employ the various vortex turbulence decay functions.
>
> Vehicle Kinetics - Contains the functionality to provide movement of an object. In this case the Leadship's equations of motion and position would be obtained from this "Class".

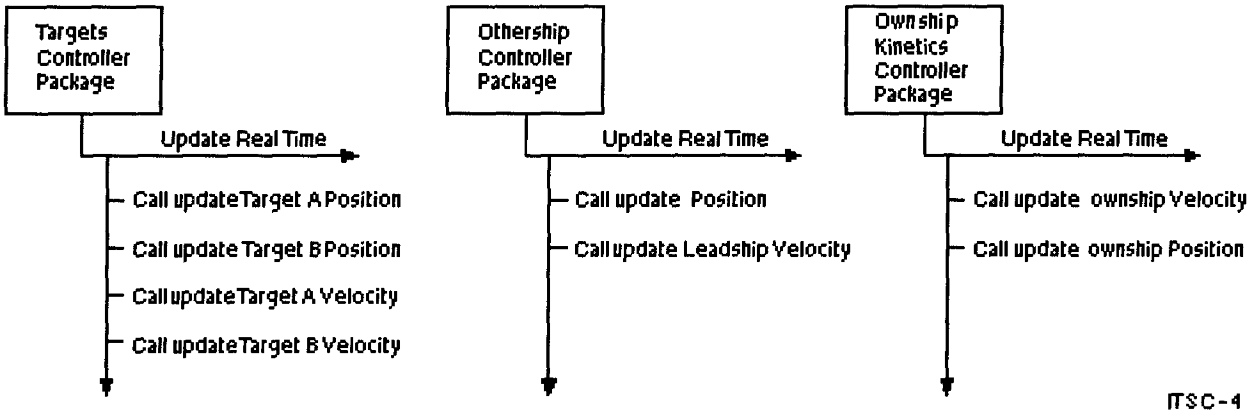


Figure 4 Example of "Subclass" Usage

Leadship Aerodynamic Interference – Contains the functionality of a disturbed air mass caused by a vehicle traveling in front of the vehicle containing the training crew. For example, the delta velocities caused by the leadship's wingtip vortex/downwash would exist in this "Class".

Lagship Aerodynamic Interference – Contains the functionality that would modify the state of the lagship attributed to the disturbed air mass generated by the leadship. The Othership subsystem would compute delta aerodynamic forces and moments from this "Classes'" functionality.

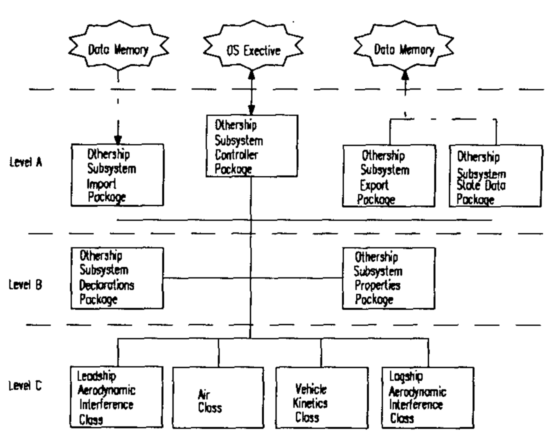


Figure 5 Second Generation OOD Othership Ada Package Structure

The packages contained in level A of Figure 5 are the only packages of this subsystem that interface to the computer OS executive. They provide the command and control for this subsystem. The design of these level A packages would be based on maximizing reuse and maintainability, while lowering complexity. Application training requirements would also have a direct bearing on the requisite fidelity for the subsystem, and therefore the composition of these packages. Package Import identifies all of the data this subsystem requires from other subsystems. The Export package defines all of the data required by other subsystems. Package State Data contains all other parameters necessary to restore this subsystem state to a previous point in the training mission. Examples of these state data parameters might be: internal computational results, previous (n-1) values, etc. The Othership Controller would still house simulation state logic, with each state being housed in an individual Ada procedure. Each simulation state would contain the appropriate model and model component control logic. This would also include the order of execution of the various compilation units.

Level B's Properties and Declaration packages need to be populated with the appropriate values for a particular application. These packages are used to make a specific instance of the level C "Classes" from the software reuse library. It is the goal of these level B packages to maximize reuseability. The Properties package contains the values of the physical and behavioral data (e.g., wing span, power rating, etc.) as well as the initialization of the adjustment constants. The parameters/constants to be populated is identified by the "Classes" being utilized. The Declaration package defines and initializes all of the necessary parameters to describe the subsystem's state at any point in time. This includes all simulation states too. Externally required exports (e.g., Delta Aero Forces and Moments) and state data (e.g., airspeed, bank angle) are a subset of the parameters contained within the Declaration package. The contents of this package must consider the requirements of "Classes" being utilized.

Level C represents a software reuse library, which contains all the "Classes" required by a particular subsystem. It is this "Class" that represents the tangible engineering product. A "Class" will contain generic models that become specific Objects when the Properties and Declaration packages are populated for a particular application (instance). For example the Lagship Bow Wave functionality for a B-2 is a specific instance of a "subclass" within the Lagship Aerodynamic Interference Class.

The major external interfaces for this second generation OOD Othership subsystem is shown in Figure 6. The primary difference is the introduction of the Mission Generation and Instructor Station subsystems. These subsystems would provide flight path commands (e.g. heading, airspeed) and vehicle configuration such as gross weight. The Visual Interface subsystem remains unchanged from Figure 2. The Targets subsystem interfaces were no longer required for the new Othership subsystem. Preliminary analysis of these interfaces indicated that the total number of interfaces did not significantly change due to this partitioning.

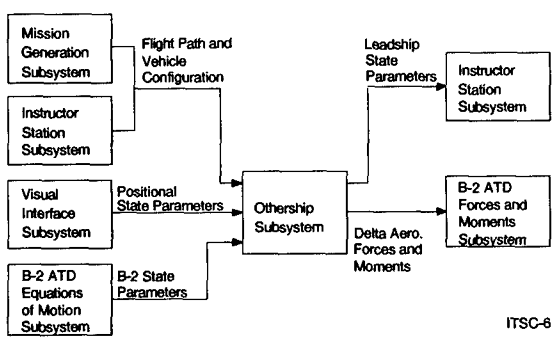


Figure 6 Second Generation Major External Interfaces

The second generation Othership subsystem has not been implemented on the B-2 ATD. This would require the B-2 ATD to reconfigure to the advanced partitioning scheme and utilize the software reuse library. This is not possible under current program constraints.

LESSONS LEARNED

The ability of the B-2 ATD's Othership subsystem to be customized quickly and at low risk for performance related changes and training requirement changes, can be attributed to several items. Most significant is the second generation OOD structure using an advanced partitioning scheme. Other items reinforce the lessons learned in ref. 1.

Four primary lessons were learned during the customizing process of the original Othership subsystem. They are as follows:

1. The second generation of the Othership subsystem has improved reuseability and maintainability because it uses the software reuse library "Classes". These "Classes" are proven pre-tested products allowing for reduced design, code and test schedule when assembling the subsystem. Regression testing is minimized because "Classes" contain proven equations in a generic engineering format.

2. The original Leadship object represented the unique attributes and functionality of a flying platform creating a disturbance in the air mass. It lacks the control of the leadship motion that exists within the Targets subsystem. The second generation Othership will assume the Leadship kinetics placing all leadship attributes related to aerodynamic interference into a single subsystem. This would enhance reuseability of the Othership subsystem and allow for quicker response to high level testing and subjective inputs. It would also reduce duplication of Leadship specific parameters.

3. Isolation of models and model components was accomplished through the use of logic within the Controller. This isolation assured that only the appropriate software units (e.g. bow wave fade in/out function) were touched to address areas where performance needed improvement. This feature also allowed for quick response to changes in the subsystem's functionality and training requirements.

4. Generic engineering notation, which includes adjustment constants, reduced the time required for high level testing, regression testing and subjective evaluations.

CONCLUSIONS AND RECOMMENDATIONS

The conclusions and recommendations are based on customizing of the original OOD Othership subsystem. Customizing was necessary to accommodate changed training requirements, results of integrated system level testing and subjective evaluations. The following is listed in order of importance:

The second generation OOD structure of the Othership subsystem addressed all changes encountered during the customizing process. It has increased reuseability and maintainability while reducing schedule requirements because it utilizes pre-tested generic "Classes" from a software reuse library.

The Othership subsystem should be enhanced to include control of the leadship motion. This would centralize all of the aerodynamic interference leadship attributes into a single subsystem.

The original OOD structure adapted very well to a changed training requirement, removal of functionality and certain performance changes. It did not resolve all of the problems encountered because of the older partitioning scheme used to create the original subsystem.

REFERENCES

1. Weiss, Jerome M., Korba, Ruth E. "Advantages of an Object-Oriented Design Approach to the Simulation of Leadship Effects", I/ITSEC Nov. 1991.

2. Weiss, Jerome M., Korba, Ruth E. "Othership System, Software Detailed Design Document", Link SDDD BD101-0102- 02, May 1994.

3. Flynn, Thomas, Petryszyn, Mary "An Analysis of Ada, Object Oriented Design, and Structure Modeling as Implemented in a Moving Target Simulation Design", I/ITSEC Nov. 1992.

MEGAPROGRAMMING AND METHODS OF REUSE: THE NAVY/STARS PILOT PROJECT

Brian E. Cahill
DUAL Incorporated
Lake Mary, Florida

Constance N. Lambert
Naval Air Warfare Center Training Systems Division
Orlando, Florida

ABSTRACT

Many software organizations have not adopted software development practices that foster reuse in any formal manner. As the simulation and training industry moves into the twenty–first century, these organizations must evolve or they will become less and less effective in an increasingly competitive marketplace. A reuse strategy is invaluable as a method of risk reduction. There are five levels of risk reduction based upon degrees of reuse.

An organization that has no formal organizational or project–level reuse strategy generally does accomplish some unmeasurable amount of *ad hoc* reuse. A common example of this is when an engineer has to provide certain functionality, and they reach into their "bag of tricks" and pull out a piece of code from another application, possibly in another language. The lowest level of quantifiable reuse–based risk management is *opportunistic* reuse, which is implemented at a project level, making use of some automated tools, with little or no unifying direction from the organization. The next degree of risk reduction is *integrated* reuse, in which the organization has adopted some form of reuse strategy, which is used consistently throughout the organization. The fourth level is *leveraged* reuse, which adopts a product line philosophy and integrates reuse tools with the software development environment. The software engineer recognizes commonalities and variabilities in their current design task within the product family, and creates a design that reflects those elements, anticipating future reuse of the code. The highest form of reuse–based risk management is *anticipated* reuse, in which the organization pursues new business opportunities that take advantage of the organization's reusable assets, as well as opportunities that will further develop the product line.

On the Navy/STARS pilot project, using the process–driven, two life–cycle approach of megaprogramming, the strategy of choice was leveraged reuse. This paper outlines the various methods of creating reusable code, as well as the structural and environmental considerations that can make reuse an attainable goal or a sizable effort. It also addresses the experience gained and lessons learned in fulfilling the concepts of leveraged reuse on the Navy/STARS pilot project.

ABOUT THE AUTHORS

Brian E. Cahill is a software engineer for the Simulation and Training Division of DUAL Incorporated, where he is currently performing domain engineering on the Navy/STARS Demonstration Project. He has five years of experience within the simulation and training field, with experience in maintenance training, cockpit procedures trainers, and celestial navigation trainers. Mr. Cahill holds a B.S.E. degree in Computer Engineering from the University of Central Florida.

Constance N. Lambert is a project engineer for the Naval Air Warfare Center Training Systems Division, where she is currently performing project support engineering on the Navy/STARS Demonstration Project. She has three years of project engineering experience from working with the Harpoon and SLAM missile systems. Mrs. Lambert holds a B.S.E. degree in Aerospace Engineering from West Virginia University.

MEGAPROGRAMMING AND METHODS OF REUSE: THE NAVY/STARS PILOT PROJECT

Brian E. Cahill
DUAL Incorporated
Lake Mary, Florida

Constance N. Lambert
Naval Air Warfare Center Training Systems Division
Orlando, Florida

INTRODUCTION

STARS

STARS (Software Technology for Adaptable, Reliable Systems) is a long-term Advanced Research Projects Agency (ARPA) project aimed at advancing the management, quality, adaptability, and reliability of Department of Defense (DoD) software intensive systems. Over the years, the STARS project has gradually focused on enabling a paradigm shift of DoD software practices to *megaprogramming.* [Boehm92]

Megaprogramming

The central megaprogramming concept is a process-driven, two life-cycle approach to software development. One life-cycle spans the creation and enrichment of an organization's capabilities for a family of related products, or *domain*. The other life-cycle spans the construction and delivery of individual products to customers, or *instances* from the domain. Such an approach can provide substantial opportunity for *leveraged reuse*, that is, planned use of adapted software components in multiple products.

Navy/STARS

Much of the STARS effort to date has been directed toward the development of tools and processes to support megaprogramming. The STARS project is now in a transition and demonstration phase aimed at supporting the transition to institutionalized usage of megaprogramming in the DoD and the supporting industrial base. There are demonstration projects under way in three services, Air Force, Army, and Navy.

The Navy/STARS demonstration project is jointly sponsored by ARPA and Naval Air Systems Command (NAVAIR). The Naval Information Systems Management Center (NISMC) is responsible for transitioning the Navy/STARS project experiences into conforming standards. The primary organizations involved in the execution of the project are the Naval Air Warfare Center Training Systems Division, the Boeing Company, DUAL Incorporated, and Enzian Technology Incorporated.

Navy/STARS is using a process developed by the Virginia Center of Excellence (VCOE) called the Reuse-Driven Software Process (RSP), also known as Synthesis. The purpose of Synthesis is defining a domain, specifying instances from the domain, and implementing designs that leverage whole and adapted components for reuse in new systems. Synthesis involves the definition, analysis, specification, and implementation of a domain which encompasses a viable product family — a family which shares sufficient commonality (and predictable variability) to justify an investment in the domain. Individual products are developed as instances of the domain, which reuse common elements of the domain and adapt variable elements using a defined, repeatable process. Synthesis defines the effort associated with creating the reusable, adaptable assets as *domain engineering*, and the effort in creating specific products as *application engineering* (see Figure 1). [SPCRSP93]

The Navy/STARS demonstration project is in the domain of simulator-based training, specifically the U.S. Navy's domain of Air Vehicle Training Systems (AVTS). If megaprogramming proves useful in this domain, it promises dramatic increases in productivity and quality, as well as corresponding reductions in the cost of building simulations.

A vertical slice of the AVTS domain was selected for a pilot project effort. This was done in order to gain experience in the Synthesis process and create and

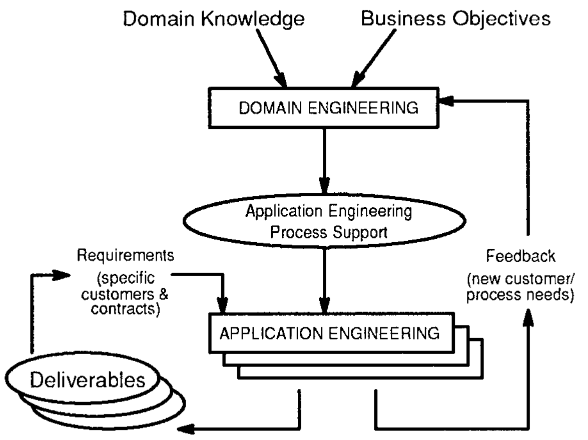


Figure 1. The Synthesis Process

refine domain–specific guidelines and methodologies before attempting full–scale domain engineering. The selected slice was very narrow, with full depth, consisting of an adaptable simulation of the TACAN and VOR components of the Navigation/ Communication subdomain of AVTS. Taking this slice of the domain allowed for focus on process vice product. As a result, every activity in the Synthesis process was performed at least once in the pilot effort.

REUSE AS A RISK REDUCER

As the simulation and training industry moves toward the twenty–first century, the costs, schedules, and related risks to developing software intensive systems are increasing at an alarming rate. For software–based organizations to survive, they will need to adopt methods to reduce the risks involved with software development. One major method of mitigating these risks is the reuse of current assets. The motivation to implement a software reuse program includes increased productivity, increased responsiveness to customer needs, improved quality, early requirements verification, reduced new development, and retained and leveraged technical expertise.

But what exactly is reuse? There is often quite a difference between an asset that is reused and one that is reusable. [Tracz88] Some definitions of reuse are quite narrow, such as "Reuse is the reapplication of code," or "Reuse is the use of subroutine libraries." Because these definitions center around the reapplication of code, and software languages generally induce some amount of specificity, reusable code fragments in this context tend to be very small. Building software systems out of these small components usually requires much work in the area of the architectural superstructure that binds all of the components together. The added cost of building and testing this architectural superstructure often outweighs the cost savings of reusing the smaller components. As a result, the more narrowly defined views of reuse have rarely shown much return on investment. As candidate components get larger and larger, however, they tend to become more and more specific. This reduces the likelihood of being a suitable candidate for reuse, because the possibility of the same set of requirements coming along is quite small. [Biggerstaff89]

Some of the broader definitions of reuse are not very practical, either. Some consider that during maintenance, the maintenance engineer is continually reusing the whole infrastructure of the system being maintained. Others consider that subprograms are reused whenever they are called at execution time. Such broad definitions help to increase reuse percentage estimates on periodic status reports to upper management, but do not provide any realistic, tangible increase in the productivity of software engineers. It's just "business as usual."

For the purposes of this discussion, reuse is practically defined as any procedure that produces (or helps produce) a system by reusing an asset from a previous development effort. These assets may either be adapted or not adapted, to solve varying problems, where adaptation is the process of modifying the asset to meet a particular requirement.

The VCOE, in its Reuse Capability Model (RCM), defines four quantifiable stages in the risk reduction growth implementation model: opportunistic, integrated, leveraged, and anticipated. [SPCRAG93] A fifth stage, ad hoc reuse, has been occurring since the earliest applications of computer programming. Ad hoc reuse has no defined reuse process and its benefits are difficult to measure.

Ad Hoc Reuse
Many organizations with no formal reuse strategy participate in the method of *ad hoc* reuse. This approach is performed informally by individuals on particular projects. It is not part of the organization's development process.

An example of an engineer using this method is when they are requested to provide a simulation of a particular device with specific functionality in the Ada programming language. The engineer may review previously developed work products. If a similar unit is discovered that is written in Ada, some or all of it may be reused in the new simulation. If a similar unit is found that is written in a different programming language, the engineer may still use the unit as a guide in meeting the current task requirements. The benefits of this type of reuse are at best difficult to measure, since the planning and recording of the reuse is virtually nonexistent.

Opportunistic Reuse
The second level of reuse–based risk management, and the first that provides quantifiable return on investment, is *opportunistic* reuse. This method requires the development of a reuse strategy for an individual project. However, the reuse is not yet supported in the organization's standard development processes. Project software plans reduce risk by defining possible areas for reuse and determining where the reusable assets may be located. At this level, specialized reuse tools, manual or automated, may be introduced into the development process. Current development needs govern the reused assets, which can range from requirements to coded software units. Risk is reduced by targeting particular areas for reuse and limiting where the reusable assets are located. This method provides a basis for measuring the benefits realized, provided reuse data collection is an element of the development process approach.

Integrated Reuse
Integrated reuse provides the next degree of risk reduction, in which a standard reuse process strategy is integrated into the organization's standard development processes. The organization's policies and procedures are structured to support these processes. There is full participation throughout the organization in developing the standard processes and tailoring the reuse tools. Asset commonalities among current requirements are identified and used as the basis for development of adaptable assets. Multiple projects may use these assets for similar needs with adaptation. Risk is reduced across various projects since the targeted reusable assets are developed for many uses, thereby reducing the new development requirements.

Leveraged Reuse
The fourth level is *leveraged* reuse, which adopts a product line strategy. The product line is comprised of specific instances of a product family with similar needs and requirements. This strategy takes into account how the reusable assets can benefit the related products in the product line.

Adaptable assets developed for the current project requirements include the commonalities and variabilities for many instances of the product line, as related to the current project requirements. These variabilities and commonalities are often interrelated, dependent upon, and embedded within each other. An example of this is an engineer developing an instance from a family of devices, which may or may not have a self diagnostic feature. This self diagnostic feature runs an automatic check on the system to ensure there are no problems. One high level variability of the system is to determine whether the desired unit has the self diagnostic feature. If the unit does require this feature, a commonality that may come forth is a power loop test, with a related variability in the other specific functions of the unit tested. The detailed system development is completed for the current requirement and additional variabilities are identified and scheduled for inclusion in future development plans.

Also at the leveraged reuse level, a software development environment is introduced with integrated reuse tools and formal processes for adapting the reusable assets. This method allows risk reduction by avoiding redeveloping areas of commonalities, improving quality as the assets mature, retained and leveraged technical expertise, and earlier validation of common requirements.

Anticipated Reuse
The highest level of reuse–based risk reduction is *anticipated* reuse, in which future customers' needs

are anticipated when reusable assets are developed. New business opportunities are pursued based on the possible application of current reusable assets and the opportunity to further develop and mature the product line and its related processes. These processes should also be readily adaptable to other product lines. The assets chosen for adaptation are in areas of higher payoff due to high complexity, higher redevelopment costs, and relative commonality. The risks are further reduced with an increased responsiveness to the customers' needs since many assets are developed with future requirements identified.

METHODS

Composition–based Systems

The composition–based model of reuse is based on the notion of plugging components together, with little or no modification to those components, in order to create target software systems. [Biggerstaff89] In theory, composition–based systems emphasize templates and abstract algorithms based upon abstract data types. In practice, composition–based systems emphasize information hiding, and the treatment of software components as black boxes, with clearly defined interface specifications. The internal workings of these components are viewed as being some unknown "software magic." This is the traditional view of software library systems, which have been in use in one form or another since the earliest applications of software systems.

Generation–based Systems

The generation–based system is aimed at reusing patterns that drive the creation of specific or customized versions of themselves. [Biggerstaff89] The primary parts of the software system that are consistently reused in generation–based systems are the architectural structures. Generation–based systems fall into three main categories: language–based systems, application generators, and transformational–based systems.

Language–based Systems. Language–based systems emphasize their well defined specification languages. These systems often look like compilers. They represent a problem domain, and hide the details of implementation from the user.

Application Generators. Application generators have no single point of emphasis, but instances of these systems always share a common architectural pattern, which is embedded in the generators' design.

Transformational–based Systems. Transformational–based systems emphasize formalization of processes that allow the generation of multiple instances of a family of systems. They focus upon the role, structure, and operation of transformations in the evolution of high–level specifications into operational programs.

Architectural Considerations

One must recognize the need for standards that transcend any component or set of components, if any reuse strategy is to succeed. These standards must apply to the data that is interchanged between components, as well as the architectural standards that impose structural patterns on systems. Broad domain standards are essential for the coordination of sets of software components so that they can be grouped based upon their functionality, inputs, and outputs. There is a direct relationship between such standards and the notion of an architecture that transcends single components. A set of components in a library, in order to be suitable candidates for reuse, must be designed with the same or, at the very least, similar architectural considerations that reflect the nature of the problem space, as well as the computational needs of the system. [Biggerstaff89]

What qualities are exhibited by a good architecture? At the very least, an architecture that is designed to promote reuse must exhibit two basic characteristics. First, it must provide a partitioning strategy, so that every component has a logical "home". Second, it must have a clearly defined communication scheme. After all, the best–designed component interface specifications are useless unless the architectural superstructure in which the components reside has well established lines of communication.

Environmental Considerations

The Ada programming language was designed to support the development of large programs composed of reusable software components. Some of the features of Ada that support reusability are:

1) Ada provides an ample variety of program unit types with syntactic interface specifications,

2) Separation of interface specifications and bodies provides information hiding capability,

3) Strong typing allows consistency between definitions and actual parameter calls,

4) Generics allow reusable uniformities of a family of software components to be captured by a single generic definition,

5) Program libraries with separately compiled program units foster modular designs. [Wegner87]

The establishment of a programming language that supports the principles of reusable software engineering was a good first step, but processes and environments are also a necessity for any reuse effort to be successful. As mentioned earlier, code libraries have been in use since the early days of programming, but we cannot expect all of our needs for building and modifying software to be met by a simple library of components. Any software engineering system that is to foster reuse must provide powerful techniques for interconnecting and modifying components as well as powerful cataloging and retrieval facilities. [Goguen87] Beyond software components and code fragments, though, there are other software engineering assets that can show considerable benefits through reuse. A reuse-supportive software engineering system should therefore also provide the capability to store and retrieve at a minimum, requirements, specifications, and documents, as well as code.

PILOT PROJECT EXPERIENCE

Architecture

The software architecture implemented on the Navy/STARS pilot project is the Domain Architecture for Reuse in Training Systems (DARTS), developed by Boeing Huntsville. DARTS is a derivative of the Modular Simulator (Mod Sim) architecture and the Software Engineering Institute's (SEI's) Air Vehicle Structural Model (AVSM).

The DARTS architecture is based upon a generic flight simulator, partitioned into twelve logical segments (see Figure 2). Segments are characterized by being internally coherent and loosely coupled externally. Interfaces between segments are clearly defined and all data transfer between segments is handled via message passing at the segment executive level, through a "virtual" network. This virtual network, which may be entirely virtual or actually a physical network, affords the architecture a high degree of flexibility. The segments can be grouped in any order or number, on one to twelve

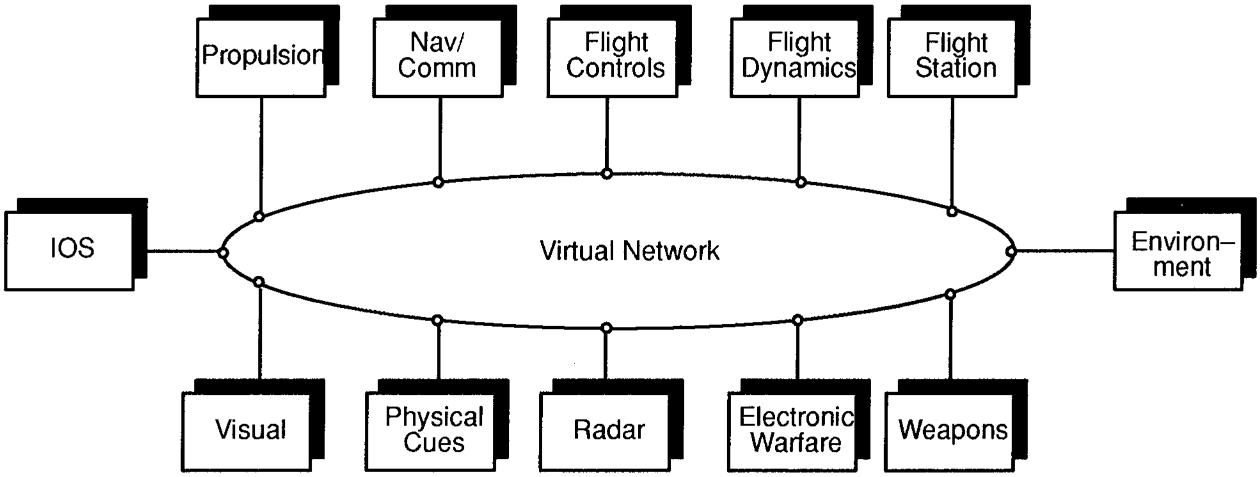


Figure 2. DARTS Architecture

computational systems, as the size of the application requires. [Crispen93]

Environment

The Software Engineering Environment (SEE) in use on the Navy/STARS pilot/demonstration projects has been developed by Boeing Seattle, teamed with Digital Equipment Corporation (DEC). It provides high leverage automation for the STARS reuse, process and technology support concepts. It is built on a foundation of commercial hardware and software products, as well as the Boeing–developed Reusable Object Access Management System (ROAMS), which provides processes for adapting AVTS assets to the needs of specific requirements. The SEE also features its own process control language, AAA (Agents, Artifacts, and Activities), which supports the object–oriented repository–based, process–driven development approach. The SEE generates a set of adapted reports including system requirements, application model, decision map, and source code analysis. The SEE also provides means for development of both domain engineering and application engineering work products, including the implementation of a precedence network, metrics collection, and problem correction. See Figure 3 for a functional model of the Navy/STARS SEE.

Reuse and Adaptation

The Navy/STARS approach can be best described as a transformational–based approach. In fulfilling the concepts of leveraged reuse using the Synthesis process, the first step was to define a domain. As stated earlier, the chosen domain was the U.S. Navy's domain of Air Vehicle Training Systems (AVTS). The DARTS architecture was well–suited to the AVTS domain. And the flexibility of the DARTS architecture allowed for certain segments unnecessary to the AVTS domain (e.g., weapons) to be adapted out of the system.

The strategy that megaprogramming pursues in the domain engineering life–cycle is to capture domain expertise in a set of reusable, adaptable assets that represent a family of systems, rather than trying to reuse existing assets with high degrees of specificity in the traditional manner. On the pilot project, domain experts were used to identify commonalities and variabilities across the AVTS domain, specifically within the T–series aircraft (T–34, T–44, and T–45). Following the Synthesis process, an adaptable simulation was produced, as well as processes for application engineering, most notably a decision model.

In an application engineering life–cycle, specific customer requirements are evaluated, and the decision model is executed, capturing those require-

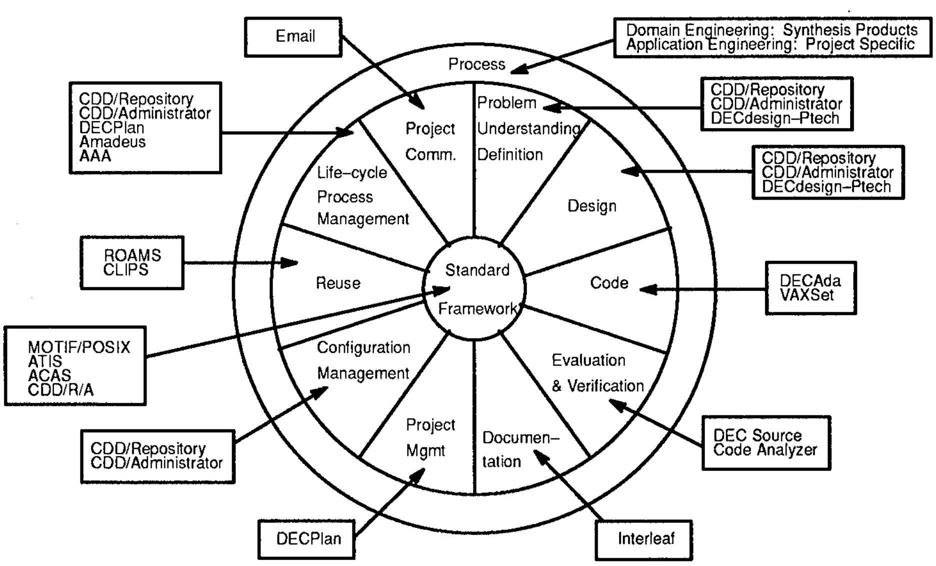


Figure 3. Navy/STARS SEE

ments for use in adapting the reusable assets. The application engineer performs detailed requirements analysis, and researches the desired target application. Once a clear definition of the target application has been made, the application engineer sits down at the SEE and answers logical groups of questions from the decision model, known as decision groups.

The answers that the application engineer gives to questions presented in the decision model are used to generate values for *instantiation parameters*, which are in turn used to instantiate the reusable assets into specific products.

An example of how instantiation parameters are used to adapt the reusable software follows:

```
--$IF $P_TACAN_SELF_TEST$ THEN
   TACAN_Self_Test;
--$END IF
```

In this example, the instantiation parameter is P_TACAN_SELF_TEST. It is used within a metalanguage (lines that are prefixed by '– –$') embedded in the adaptable Ada source code file. While executing the decision model, the application engineer is asked "Does this TACAN radio have a self test feature?" The answer is used to assign a value to the instantiation parameter, which is in turn used when the adaptable code is retrieved and adapted to meet the current requirements. If the answer is yes, the metalanguage is stripped out, and the call to the TACAN_Self_Test procedure is present in the adapted code. If the answer is no, the metalanguage and the Ada code within the conditional metalanguage structure are stripped out, and no call to the TACAN_Self_Test procedure will be found in the adapted code.

More detailed questions follow the question of feature existence described above, including questions regarding the phases of self test (duration, indications, method of initiation, etc.), and its effects on other components in the system. The code fragment above demonstrated an inclusion/exclusion use of instantiation parameters. Substitutions are also supported, as in:

```
NUMPHASES : integer := $P_Num_Phas$;
```

In this example, the application engineer is asked "How many phases are there in this self test?" They answer with an integer response, '2' for example, and when the code is retrieved and adapted, the line of Ada becomes:

```
NUMPHASES : integer := 2;
```

Having the capability to adapt any and every line of code in a system provides great leverage to the reuse effort. It is extremely important, however, to carefully analyze and manage the level of adaptation. This vigilance will ensure that the time and effort spent in maintaining and testing the adaptable system do not outweigh the savings afforded by the reuse potential.

Lessons Learned

The pilot project effort proved invaluable as a learning experience. A general lesson learned was that Synthesis is is a difficult process to learn without practical experience. A small, well–defined pilot effort can be of great use in learning the process before attempting full–scale domain engineering.

The one factor that is essential for reuse to succeed on a large scale is domain knowledge. When considering candidate assets for reuse, one first thinks of the widely applicable functions (sorts, searches, stack, string, and math operations), but they only represent a small fraction of most large scale applications. The vast majority of code lies in the domain–specific portion of software systems. Expertise and in–depth knowledge of the domain allows for the maximum exploitation of reuse opportunity.

Analysis of variability and commonality results in generic adaptable architectural components that would not be created had a single point approach been taken.

Use of metalanguage within components to implement adaptability breaks down the barriers that a generic Ada component provides. Metalanguage–based adaptability provides almost unlimited flexibility.

Automation of decision models to adapt code can be complex and costly. Careful metrics must/will be collected in the next phase to provide feasibility information.

Commonalities and variabilities associated with this experience only involved the T–34, T–44, and T–45 aircraft. As other aircraft are considered, the domain complexity will significantly increase during each iteration through the Synthesis process.

Defining the domain (especially commonalities and variabilities) is time consuming, but with the right documentation and personnel experience, the process leads to many benefits.

The experience using the TACAN and VOR brought to light commonalities and variabilities throughout AVTS.

It is important that the decision model capture and group the choices between variations of domain instances according to some logical scheme (e.g., instead of a large decision group called 'TACAN decision group', we generated smaller logical groupings of decisions, like 'Power', 'Self Test', 'Outputs', etc.).

It is equally important to organize decision groups in a hierarchical structure to capture dependencies, in order to realistically model the real world (e.g., if the TACAN does not exist in an instance, there is no need to execute the other TACAN decision groups).

Synthesis requires identification of software components and their hierarchical relationship (product architecture) before beginning component design. This emphasis on careful partitioning of the system is similar to that found in object–oriented design.

Synthesis tends to be life–cycle oriented rather than technique oriented. Therefore, traditional tools such as structural analysis, object–oriented analysis, program design language (PDL), pseudo–code, etc. were useful techniques for accomplishing the goals of Synthesis.

CONCLUSION

Reusable software engineering is essential for a modern software engineering house to survive. Simple code libraries are a thing of the past. Complete software engineering environments devoted to reusable software engineering techniques and principles will facilitate the reusable software engineering revolution. The key to maximizing the benefits of reuse is in domain analysis. Only with sufficient understanding of a well defined domain can truly reusable, adaptable assets be created.

At this time, the Navy/STARS project is capable of demonstrating a simple application engineering session. That is, we can demonstrate the ability to enact an instance from the AVTS domain.

The results thus far have been encouraging. Careful definition of the scope of the sub–domains has increased our understanding of the underlying relationships between domain instances, which has had the effect of increasing our level of reuse without further effort. We have seen that the increased investment in design and test required for adaptable components has resulted in higher quality components. One surprising discovery has been the degree to which adaptable documents have been easy to create while being a valuable reusable component.

At the time of this writing, the pilot project had completed a successful readiness review and a demonstration project planning phase, wherein guidelines and methodologies were refined and domain engineering strategies were developed. The team is proceeding with the demonstration project, performing full–scale domain engineering. There is a demonstration milestone in the fall of 1995, after which DUAL Incorporated will begin an application engineering cycle, and produce the first instance of this domain, a T–34C Flight Instrumentation Trainer scheduled for turnover to the fleet in 11/96.

REFERENCES

[Biggerstaff89] Biggerstaff, T. and A. Perlis. *Software Reusability.* 1989. ACM Press, New York, NY.

[Boehm92] Boehm, B. and B. Scherlis. "Megaprogramming", *Proceedings of the DARPA Software Technology Conference.* 1992. Meridien Corporation. Arlington, VA.

[Crispen93] Crispen, R., et al. "DARTS: A Domain Architecture for Reuse in Training Systems". 1993. *15th I/ITSEC Proceedings.*

[Goguen87] Goguen, J. "Reusing and Interconnecting Software Components", *Software Reusability.* 1987. Computer Society Press of the IEEE. Washington, D.C.

[SPCRAG93] *Reuse Adoption Guidebook.* 1993. SPC–92051–CMC. Software Productivity Consortium Services Corporation. Herndon, VA.

[SPCRSP93] *Reuse–Driven Software Processes Guidebook.* 1993. SPC–92019–CMC. Software Productivity Consortium Services Corporation. Herndon, VA.

[Tracz88] Tracz, W. *Software Reuse: Emerging Technology.* 1988. Computer Society Press of the IEEE. Washington, D.C.

[Wegner87] Wegner, P. "Varieties of Reusability", *Software Reusability.* 1987. Computer Society Press of the IEEE. Washington, D.C.

WEAPONS SIMULATION EXECUTION, IN THE TARGET? OR IN THE SHOOTER?

Ted Clowes
Cubic Defense Systems, Inc.
San Diego, California

Abstract: As the DIS community moves towards incorporating live training range players into their games, a number of issues arise. This paper provides background on the set of problems unique to the range community and addresses them relative to the issue of weapon simulations and whether their execution should be based at the target or within the shooter. The issues addressed include: low communications bandwidth compared to simulators; intermittent communications paths or dropouts; available processing power; and classification. These issues are primarily related to live training ranges that impose real-time and real-world constraints. Since it is desirable to have simulators provide pseudo threats and players that interact with real players, it is necessary to understand these constraints.

Examination of some Army, Air Force, and Navy ranges and their restrictions relative to rate of player communication, and amount of data that can be passed is presented. This is contrasted with the typical capability of simulators. The effects of communication dropouts or path unreliability is then added. Some ranges and types of players are less susceptible to this problem than others. Next is a brief discussion of the class of processing power available at the player unit and the restrictions this imposes on the approach to simulation execution. Some time is also spent on the issue of classification and the problems that are introduced when you want to use classified weapons models in a world that is inherently easy to monitor. The conclusion presents a recommendation about where the weapons simulations should be executed when dealing with live ranges and a mix of real and pseudo players.

Author: Ted Clowes, Staff Scientist for Cubic Defense Systems, Inc. (619)277-6780. Mr. Clowes has a BA in Physics and Mathematics from Western Washington State and an MS in Computer Science from the University of California at San Diego. His primary orientation, since the mid 70's, has been in the instrumentation and real-time tracking of moving objects. He is currently working in the training ranges arena, with emphasis on airborne units.

WEAPONS SIMULATION EXECUTION, IN THE TARGET? OR IN THE SHOOTER?

Ted Clowes
Cubic Defense Systems, Inc.
San Diego, California

INTRODUCTION TO THE RANGE WORLD

The range world encompasses the "live", portions of the "virtual", and small portions of the "constructive" simulation domains. It deals primarily with instrumented player units that have a degree of free play, while training in an actual environment. The function of the range is typically to provide a safe, monitored environment that allows a class of live action to occur for the purpose of training. Depending upon the type of range, there may or may not be real weapon or threat events combined with simulated versions of these events. In some cases, there will be referees on the range, while in others remote instructors will be monitoring the activities. Ranges can allow instructor interaction with the players in real time or depend solely on After Action Reviews to instruct. Combinations between these extremes are possible as well. Ranges cover air, land, sea, and underwater. They vary in size from a couple of kilometers on a side (Snort) to thousands of kilometers on a side (Vandenburg). The entities that are tracked fall into classes that tend to be based on maneuverability. Example classes are described below.

Slow Movers (e.g. dismounted players)

Typically a slow mover is an individual on foot, usually a combatant. The characteristics of a slow mover include: low top velocity, typically less than 15 kph; small target size, typically less than 2 square meters; limited armament capability unless resupplied; limited transportable defensive capability; high degree of unpredictability; and usually acts as part of a group. There are other possibilities for slow movers besides infantry. These can include balloons, submarines, divers, field artillery, and paratroopers; however, most of these stretch at least one of the criteria, usually target size. The problems associated with a slow mover are: tendency to be non-cooperative from a communications perspective; limited load carrying capability for instrumentation; and difficulty of providing the player with convincing simulated inputs (e.g. environmental conditions such as fog, dust, rain, or smoke). This class of entity, when on a range, is updated or interrogated at low rates. These rates are fractions of a hertz or multiple second intervals.

Medium Movers (e.g. tracked vehicles)

Medium movers have a vast variety of capabilities and are not represented by a typical constituent, as is the slow mover. Some of the more unusual slow movers, (e.g. submarines) are capable of operating in the class, but tend not to. The characteristics of this class include: velocities less than 200 kph; target sizes large enough to be seen from a distance, when not camouflaged; medium to heavy armament capability without resupply; capable of extended cruising range larger than the typical range tracking area; transportable defense capability, either in the form of armor, maneuverability, or ability to take damage and continue to operate; reasonable amount of predictability in maneuvers; have their own local power source; have some amount of excess load carrying capability; and the ability to operate effectively alone or as part of a larger group. The possible members of this group are: armored vehicles; surface ships; helicopters; gliders; low-performance aircraft; remotely-piloted vehicles; supply vehicles; and in some circumstances, underwater vehicles. The problems associated with a medium mover are: its size and range tend to require a larger playing area than is available without artificially restricting its movement; difficulty of simulating logistics resupply problems without artificially restricting the player or extending the duration of the exercise significantly; and inconsistency between platforms and their interfaces that need to be monitored or stimulated. This class of entity, when on a range, is updated at low to medium rates. These rates rarely exceed one hertz.

Fast Movers (e.g. aircraft)

A typical fast mover is airborne with characteristics which include: velocities greater than 200 Kph; target sizes large enough to be seen from a distance; medium to heavy armament capability; capable of traversing a substantial portion of the range area in much less than the duration of the exercise; heavy defensive capability sacrificed to achieve speed; highly maneuverable, but still predictable; have their own local power source; have some excess load carrying capability; and tend to operate alone, though can coordinate with a small group. The possible members of this class are:

high-performance aircraft; missiles; high-performance, remotely-piloted vehicles; and in some instances, helicopters. The problems associated with this class include: the ratio of time necessary to set up an engagement to the actual engagement time during an exercise is large and costly, due to the speed issue; maintaining sufficient communication connectivity to accurately predict maneuvers; and the severe environmental constraints placed on instrumentation used to monitor the player. This class of entity, when on a range, is updated at medium to high rates. A high rate in the range environment is 2.5-20 hertz, which is lower than most simulators operate at.

Movement Implications

Based upon the movement characteristics of range players, some differences arise from the simulator world. The update rates are slower, meaning that the fidelity associated with player location is lower. This impacts the ability of a simulation that is run in a shooter to accurately predict the end game. Another issue is size, the slower movers typically have a low target cross section, while the faster movers are highly maneuverable with a larger cross section. This also works against running the simulation in the shooter when combined with the fidelity of the player's location and the need to properly assess damage. Only the player really knows where its located and the associated attempts at evasion during the simulation end game.

HOW MUCH MESSAGE TRAFFIC & HOW OFTEN

One of the issues that has begun to surface in the DIS community is bandwidth. This has become more of a consideration recently as the number of players goes up. In the range world, bandwidth has always been a problem, as has the related issue of communications reliability. Most ranges must use some form of RF to communicate with the player units. These RF paths are constrained by the player types in terms of power, spectrum, weight, and cost. The result is imposed restrictions on bandwidth, path reliability, and update rate. This section provides some insight into some sample live training ranges and those restrictions.

CTC Ranges

There are currently two real-time instrumented CTC ranges, Hohenfels(CMTC) and Fort Irwin(NTC/AW) with the third, Fort Polk(JRTC) currently being upgraded to provide data in real-time. All of these ranges mix the MILES engagement simulation system with player unit tracking to provide a real-time monitored system with after-action review capability. The Fort Irwin system also provides the ability to track high-performance aircraft. The ground-based-weapons events are scored with MILES and the results are presented after the fact at the central control complex, with potential damage assessment made. These assessments are made by a combination of the MILES system, field referees and the central computer complex and can occur after the event or during the event. Message sizes vary from 20 bits to eight 16-bit words.

At CMTC, a player is only positioned every 3 seconds, and less often, if the player has crawled into a communications foxhole. This range uses multiple frequencies (FDMA) combined with time spacing (TDMA) to track up to 1100 players at once. The communication data rate is such that dropouts cause times in excess of the typical DIS heartbeat (5 seconds) for a player response to occur. These times are also long enough to complete a weapons event simulation.

At NTC/AW, the ground-player positioning system is being upgraded to permit a dynamic update rate and handle between 2000 and 4000 players. With this comes an easing of the event reporting to within 5 seconds of occurrence. The air side of the system supports up to 36 players with an update rate of at least 2.5Hz. Due to the update rate, dropouts on the air side, while they may be an annoyance, do not come close to the simulation fly-out time. Dropouts on the ground side could exceed a typical fly-out time for engaged players.

At JRTC, the ground player update rates will be similar to CMTC and NTC/AW with similar numbers of players.

TACTS/ACMI/MDS Ranges

This is the largest set of compatible range systems worldwide, which currently numbers 23 and growing. The systems, which are installed in many different countries, track and train pilots for air combat. While there are very few identical sites, the player units can be used on any site and all sites share a common communication architecture. The systems track from 4 to 36 players, depending upon the site, and have update rates from 2.5Hz to 10Hz. In general, due to the launch to eject and fall times of the weapons involved, the weapon simulations can actually run in real time even with dropouts. All of the simulations are run at a central ground based control site. Message sizes vary from 23 16-bit words to 75 16-bit words.

Others

Other training and test ranges include RMS-2, RMS-SCORE, EATS, STS, MSR, LATR, TCTS, TOAME, FORACS(undersea), and a number of others located at a variety of sites. Some of these ranges (e.g. LATR, TCTS) are not due to be operational for some time. Others have been in existence for years. What they have in common with the CTCs and the TACTS/ACMI/MDS ranges is low bandwidth and low update rates when compared to the DIS simulation community. They also regularly experience communication dropouts that are not necessarily predictable.

DROPOUTS AND PATH RELIABILITY

In the instrumented player world, communication paths are far from the near perfectly reliable item that they are in the simulation world. While portions of a range may have the same types of direct connections that are used in the simulation world, inevitably there is an RF connection to the individual player units. These radio connections are the root of any communication problems that occur in the system.

Some Reasons Why They Occur

Radio connections, by their very nature, are not perfect. In a wired connection, such as the simulation world uses, the types of potential problems usually are limited to insufficient bandwidth or broken physical connections. The bandwidth usually shows up as too high a collision rate on the ethernet (a collision detection, collision avoidance protocol), while the broken physical connection does not show up at all. While these same problems appear in the radio world, some additional difficulties also occur. Radio does not depend upon a medium to work, but it is affected by the medium it operates through. What this means is signal attenuation on rainy days, multipath over flat surfaces, or no signal (depending upon the frequency) when a tree gets in the way. Most players' primary concern is not communication reliability with the range, but rather hiding from or surprising the opposition. This orientation, particularly in ground ranges, means putting something substantial near you, which effectively cuts off communication in that direction.

To lower the incidence of these environmentally caused errors, ranges tend to use certain protocol techniques, such as Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), or combinations of these protocols. This tends to eliminate the collision issue. Retention and time-tagging of special event data until acknowledged, helps overcome the signal attenuation, multipath, and lack of signal. Some ranges also add an auto-respond feature in case the dropout is only on one side of the communication path. The result of all of this is a still less reliable communication path than exists in the simulation world.

How Often Do Dropouts Occur?

While different range designs have addressed different problems; they all experience a measurable percentage of dropouts. Ground ranges tend to have higher dropout rates than air ranges due to the more frequent opportunities for the players to shield themselves from the instrumentation. An example of this is the CMTC visibility (connectivity) testing results, which show that two-thirds to three-quarters of the players experienced greater than 95% connectivity with the overall range connectivity between 83% and 95%. This means that somewhere between one in twenty and one in six messages are lost or not completely recoverable at the instrumentation system. These numbers are averages over 2-3 hour missions, so what they don't indicate is that some portions of the range and some types of player activity are more susceptible than others to instantaneous communication loss.

On the air side (e.g. TACTS/ACMI), using a technique like auto-respond, these numbers change to an overall connectivity of greater than 95% on a properly configured range. Since the missions are shorter, typically 20-40 minutes, it is apparent that range coverage is better (not as many blanking situations) than on the ground. Communication integrity declines with altitude as terrain masking enters the equation, but the effect of dropouts is much lower. This is primarily due to the larger number of ground relay stations that are installed on air ranges.

Dropout Implications

Since communications integrity is suspect in the live environment, attempts to provide something close to real-time kill removal favor execution of the weapons simulation in the shooter. The shooter knows which target it has selected and can run the model while attempting to communicate the fire information to the target. If communications is imperfect, the real-time nature of the flyout is retained and the kill can be properly timed with respect to other events. This is important if the target is also firing at someone else. Once the event is time-tagged, then if it does not get communicated right away, it does not affect the game outcome as much.

PLAYER UNIT PROCESSING POWER

Current player units tend to be constrained by three things: power; size; and weight. These restrictions combine with the environmental constraints to force the player units into a situation where they have less computational power than the typical desktop computer. Added to this is the procurement time necessary for the DOD and, therefore, the units tend not to keep up with the electronics industry as well as the simulator equipment does.

Typical Processors

Existing player units in the CTC training range world use x86 family devices combined with TMS320 DSP devices. The DSP units are dedicated to handling the RF signals, while the control and data collection is handled by the x86 unit. While the x86 unit is similar in computational architecture to a desktop computer, it is run at lower clock frequencies to conserve power and meet environmental constraints. The unit does not support floating point operations.

Player units in the TACTS/ACMI training range world use x86 family devices, as well as 8080 family in the older units. Like the CTC units, they do not support floating point. A limited number of a special version of the units support floating point, however, all units run at lower clock frequencies than desktop units in order to meet environmental constraints.

Newer player units for a few of the range systems will include floating point and higher clock frequencies, however, they will always be slower and older than the current technology available to the ground based, environmentally controlled, simulator world.

Simulation Restrictions

As can be seen from the current player unit section, simulations to be run in the players have some constraints that affect the way they can be constructed and how they can be used. Taking a working simulation straight out of a ground system and expecting it to work in a player unit in real time is probably impractical. In addition, expectations of being able to run multiple simultaneous weapons simulations in a player unit, unless the simulations are very simple, is also impractical.

CLASSIFICATION, THE SHOW STOPPER

Classification of data, whether by the US or foreign governments, is an additional complexity that is added to the training environment. As training becomes more realistic, it provides more insight into how weapons systems work and the tactics employed to use them. Since secrecy and intelligence gathering are a basic part of war, this impacts training systems. Ideally, use of a training system provides perfect learning feedback for the trainee and no useful intelligence gathering opportunities for the unsanctioned observer. To achieve this means that the training system should not provide any classified data in the clear at a point that is capable of being monitored by an unsanctioned observer.

Kinds Of Things That Are Classified

Different organizations define what is classified differently and there is no universal guideline. However, an example of a guideline that is used for systems installed in multiple countries is OPNAVINST C5513.2B-91.1. This guideline says the problem is not the hardware, only the software and the subsequent data that might reveal algorithms or performance parameters. It further states that the type of player, most general player information, and the fact that firing has taken place is not classified. Data related to EW, avionics, aircraft, and weapons may be classified depending upon the item and the entity. Generally, data that provides information leading to performance parameters; such as sensor range, weapon effectiveness, or target selection algorithms tends to be classified. The exceptions to this are items that can be computed using a basic Physics textbook, such as ballistic projectiles.

Range Support of Classified Transmissions

A number of experiments have been performed and fielded with cipher equipment on training ranges. Some of these use standard military operational equipment, such as secure radios for audio communication. Others use special encryption engines designed for situations other than standard player tactical environment. Current operational training ranges tend to have at least some portion of the system encrypted and typically the ground system computer that replays or monitors weapons flyouts, tactics evaluation, or other sensitive information is in a secure area. The communication link to the players, in the case of real entities, tends not to contain sensitive information. If sensitive information is present, the link is encrypted or uses some sort of cipher technique. This approach in the live training community was originally used because there was insufficient computational power or data collection ability at the player level. As DIS standards are applied and the player computational capabilities go up, either the links will need to be scrubbed of sensitive information or they will need to be encrypted to protect them from the unsanctioned observer.

CONCLUSION AND RECOMMENDATION

What the real world imposes on DIS is a set of conditions that are less than perfect, either in communications reliability, processing throughput, or ability to have your sensitive data covertly monitored. As a result, simulations involving real-time players need to be less revealing of data and more tolerant of an inability to communicate what data is sent at a particular time. This can be achieved for short flight weapons (e.g. guns) or in cases of low connectivity by running the simulation exclusively in the shooter; or, for some ranges, the shooter's surrogate central ground. For longer flight weapons, simulation should start in the shooter and be handed off (e.g. by a Handover PDU) to the target, so the target can assess the effects of terminal guidance avoidance maneuvers and any subsequent damage. The contents of the hand-off message need to fit within the classification guidelines of the type of weapon being simulated or the link will need to be encrypted. With this combination of approaches, real-world entities should be able to play in the DIS world.

BIBLIOGRAPHY

Cubic Defense Systems (1993). System Specification for the Integrated Combat Maneuver Training Center Instrumentation System (CMTC-IS)/Simulated Area Weapon Effects - Radio Frequency (SAWE-RF) (CMTC A001 Rev B). San Diego, Ca. Cubic Defense Systems.

Cubic Defense Systems (1992). Software Requirements Specification for the Repeater Radio of the National Training Center Range Data Measurement Subsystem Upgrade (SRS715030). San Diego, Ca. Cubic Defense Systems.

Cubic Defense Systems (1994). Acceptance Test Report Document for the Combat Maneuver Training Center Instrumentation System (CMTC-IS) (TR CMTCA001-1A). San Diego, Ca. Cubic Defense Systems.

Institute for Simulation and Training (1994). Standard for Distributed Interactive Simulation -- Application Protocols Version 3.0 Working Draft (IST-CR-94-18). Orlando, Fla. University of Central Florida.

SRI International (1993). TACTS/ACTS Airborne Subsystems Reference Manual. Volume III. Eglin Air Force Base, Florida. Range Directorate, Training Range Programs Office, Naval Air Warfare Center-Aircraft Division and Department of the Air Force, Range and Air Base Systems Program Office, Aeronautical Systems Center.

United States Department of Commerce, National Bureau of Standards (1977). Data Encryption Standard (FIPS PUB 46). Springfield, Virginia. National Technical Information Service.

United States Department of the Navy (1988). Classification guidelines for TACTS/ACMI/MDS project (OPNAVINST C5513.2B-91.1, 4 Oct 1988).

ARPA Reconfigurable Simulator Initiative (ARSI)

Duke Buster
Jim King
Texas Instruments Incorporated
Plano, Texas

ABSTRACT

ARSI is a low cost, Distributed Interactive Simulator (DIS)-compliant simulation that can easily change shape into different vehicles. ARPA will use ARSI to explore the viability of such simulators for training and the research, development, and evaluation of future vehicle concepts. We contend that a single reconfigurable simulator will maintain the required fidelity and be less expensive than a collection of single configuration simulators.

ARSI has five areas of reconfigurability: mechanical enclosure, distribution of simulation functions, crew/vehicle interface, tactical interaction with other vehicles, and scenario / battlefield database. The keys to easy configuration are a flexible "core" from which hardware and software modules can be hung, and emphasizing the use of models whose behaviors are table-driven or parameter-driven. The baseline ARSI program will deliver this "reconfigurable core" and modules for five vehicle configurations: M1A1 Abrams, M1A2 Abrams, M2A1 Bradley, M2A2 Bradley, and HMMWV scout.

Keywords: simulator, reconfigurable, DIS, platoon training, concept development

BIOGRAPHIES

Duke Buster

Duke is the lead software engineer for ARSI. He is also responsible for the system integration and test. Duke has worked at Texas Instruments since graduating from Texas A&M in 1986. He has developed several fast prototype aircraft simulations for TI product development and man-in-the-loop testing. Duke's emphasis in these simulations has been the database and modelling functions. Outside of simulation work, Duke has developed a number of decision aids for mission planning and mission execution.

Jim King

Jim is the lead systems engineer for the ARSI program. He has worked at Texas Instruments since 1985. Prior to his work on ARSI, Jim worked on several projects involving simulation, mission planning, flight tests, and test equipment. Some of these projects include the AIWS Tactical Aircraft Mission Planning (TAMPS) interface, Covert Penetration System algorithm development and flight test, and the Thirsty Saber Flight Test Control System. Jim graduated in 1985 from the University of Texas at Dallas with a Bachelor's degree in Mathematical Sciences (computer science) with prior coursework in Electrical Engineering at Southern Methodist University and Texas Tech University.

ARPA Reconfigurable Simulator Initiative (ARSI)

Duke Buster
Jim King

WHAT IS A RECONFIGURABLE SIMULATOR?

Currently, a reconfigurable simulator is not a well defined object. People generally expect a reconfigurable simulator to be:

1) easily expanded, reduced, or altered in hardware and software components,
2) easily upgraded to new hardware and software,
3) modifiable by a documented procedure,
4) modifiable within typical resources - personnel, funding, schedule, etc.

With changes in the hardware and software, the simulator may be given a different appearance, function, performance, or cost.

Reconfigurability is a relative value. The more changes that can be made to a simulator given limited resources, the more reconfigurable the simulator. Any simulator can be modified with enough money, equipment, and personnel. To discuss reconfigurability, we have to define what pieces of a simulator can be modified and limits on the required resources.

WHY ARE RECONFIGURABLE SIMULATORS INTERESTING?

The basic reasons for considering reconfigurable simulators are:

1) the potential to save money for a group that wants more than one vehicle simulation,
2) the potential to save development time and cost for building new simulations.

Currently, a group that wants multiple vehicle simulations must develop or purchase multiple simulators. Much of the equipment between simulators is redundant, such as the enclosures and computer equipment. The group must also provide the storage space and maintenance for multiple simulators. One "good" reconfigurable simulator can save the cost of the redundant hardware (particularly expensive computer equipment) and support costs. The "good" phrase above hides the possible difficulties of a reconfigurable simulator. To be good, the simulator must maintain the required fidelity of all the vehicles, and require little effort to change shape between vehicles. A group that wants only one existing simulation would not need to consider reconfigurable simulators. The modules in a reconfigurable simulator are typically more expensive than single configuration simulator modules because the reconfiguration demands more flexibility.

For an organization developing a new simulation, a reconfigurable simulator is interesting because those modules that are common to the old and new simulations can be reused. This cuts down development costs, particularly if the reused models have already been validated and verified. Reusing a reconfigurable simulator may also shorten the development time for the new modules by providing pre-defined interfaces and functions that the designers do not have to create.

To be effective in either role (multiple vehicles or a development platform) a reconfigurable simulator must be modular. This is the basic technical challenge to making a reconfigurable simulator: maintaining the required fidelity over a range of emulations while allowing the easy addition of new modules.

HOW RECONFIGURABLE IS ARSI?

Our customer wanted a prototype reconfigurable simulator with as broad a configuration power as we could design in 18 months and the given budget. Our customer specifically requested that ARSI initially emulate five ground vehicles, provide hooks to easily expand to other vehicles and aircraft, provide a skeleton for future concept vehicle definition, and have a low recurring cost for multiple footprints. PM-CATT and Battle Labs personnel assisted us in defining the function scope for ARSI. After defining the function scope, we identified five required areas of reconfigurability:

1) Mechanical enclosure - the enclosure includes the floor, frame, seats, monitors, etc. Reconfiguring the enclosure means we can change the cockpit layout, number of crewstations, etc.
2) Distribution of simulation functions - the software modules are distributed among

several computers across a standard network. Reconfiguring the function distribution means we can change the computers on the network and put whatever software modules we wish on each computer. This also allows us to incorporate existing systems.

3) Crew/vehicle interface - this interface includes the visuals and controls provided to the crewmembers. Reconfiguring the crew/vehicle interface means we can move or change: out-the-window visuals, actual hardware controls such as grips and pedals, or panels emulated with computer displays and touchscreens.

4) Tactical interaction with other vehicles - this interaction takes place across a network to other simulations controlling the vehicles. Reconfiguring the interaction with other vehicles means ARSI can game on Distributed Interactive Simulation (DIS) and Simulation Network (SIMNET) protocol networks.

5) Scenario / battlefield database - this database(s) includes the battlefield data for the models and visuals. Reconfiguring the battlefield database means we can create and load a new gaming area from digitized databases in formats such as the Standard Simulator Database (SSDB) Interchange Format (SIF) and the Defense Mapping Agency (DMA) formats.

We set some reconfiguration goals to perform typical changes between existing vehicle configurations. These goals are listed below. Our goals assume the people making the changes are familiar with the Operators Guide or Reconfiguration Guide (manuals for changing the simulator between defined configurations or creating a new configuration, respectively). At the time of writing this paper, we have defined some reconfigurability metrics mirroring our goals but have not collected any numbers.

Mechanical enclosure:

- Assemble the simulator from stored state to a running configuration - 2 people, 1 day
- Change from one defined vehicle configuration to another - 2 people, 1/2 day
- Add or move a monitor - 1 person, 1 day (assuming the other cockpit pieces do not have to move)

Network distribution:

- Remove a machine from the network - 1 person, 1/2 day
- Add a Unix or VMS machine to the network - 1 person, 1/2 day
- Move a software module from one Unix host to another - 1 person, 1 minute
- Move a software module to a Unix machine to VMS or vice versa - 1 person, 1 hour (assuming the code is portable, this includes copying data files and recompiling)

Crew/vehicle interface:

- Modify the out-the-window visuals fields-of-view - 1 person, 1 hour
- Add a new out-the-window visual - 1 person, 1/2 day (assuming the cockpit layout is complete and the channel limit on the image generator computer is not exceeded)
- Modify an emulated controls panel - 2 people, 2 days (assuming the modelling is complete and the change is well defined)

Scenario/database:

- Switch from one defined database to another between exercises - 1 person, 1 minute
- Build a new database from digitized data - 1 person, 2 days

Although ARSI is designed for the easy addition of new modules or cockpit layouts, we have not set any reconfiguration goals for new developments. These efforts depend too much on the design complexity and the number of the designers. We will, however, collect metrics when we develop new configurations.

HOW WILL ARSI INITIALLY BE USED?

ARPA will use ARSI to explore the viability of a reconfigurable simulator for:

1) training
2) research, development, and evaluation of future vehicle concepts.

Several government and National Guard sites will experiment with crew and platoon level combat training. We are currently identifying which government agencies will work with ARSI as a new development tool. Various Battle Labs and development Commands have been discussed. The baseline ARSI program will deliver the reconfigurable core (described below) and the modules for five vehicle configurations: M1A1 Abrams, M1A2 Abrams, M2A1 Bradley, M2A2 Bradley, and HMMWV scout.

We anticipate that ARSI will become a tool for software development methodologies and environments, such as those forwarded by SEI for reuse (domain modelling), Joint Modeling and

Simulation System (J-MASS), and Software Technology for Adaptable Reliable Systems (STARS). Texas Instruments will apply ARSI within its own Integrated Product Development Process (IPDP).

HOW DOES ARSI WORK?

ARSI has two basic states - exercise and configuration. In its exercise state, ARSI works as a simple "turn it on and start training" simulation. In its configuration state, ARSI works as a toolset with a set of software and hardware building blocks. The user of the configuration state will either be defining a new database or a new vehicle/subsystem.

Throughout the design phase, we have made choices of performance versus reconfigurability. We emphasize reconfigurability as long as the simulation fidelity "satisfies training requirements". At the time of writing this paper, we have scheduled tankers to assess the configurations for training. We use existing model algorithms and data tables for the parts of ARSI that must be validated and verified: weapon models, motion models, and damage assessment models.

Exercise state

We will describe how ARSI works in the exercise state by describing the modules involved. We divide ARSI modules into these categories:

- Software: reconfigurable core, generic modules, vehicle specific modules.
- Hardware: reconfigurable core, generic modules, vehicle specific modules.

The reconfigurable core modules, both software and hardware, are the simulator skeleton. The core modules are in every configuration: simulation executive, comm process, spatial database query body, enclosure, frame, and computer network. All the other simulation modules hang on this core, so the core modules must have simple, flexible interfaces.

The generic modules provide functions common to multiple vehicle configurations. These modules are generic because they are either vehicle independent or they are table/parameter driven. The generic modules include: image generator software, scene driver, DIS/SIMNET communications process, hardware controls handler, damage assessment model, ballistic gun model, sound generator process, test/calibration process, gunsight eyepiece, radio/intercom, sound equipment, cockpit displays, and seats.

The vehicle specific modules, software and hardware, are unique to the emulated vehicle. These modules are not required to be easily ported or modified. The vehicle specific modules include hardware controls (grips), computer-emulated control panels, missile models, vehicle system models, etc.

Figure 1 shows how the software modules are connected. The software modules communicate by broadcast messages. This is the most reconfigurable design, but we recognize that it is not the most efficient. The communications process on each computer handles the message passing. Each module receives and sends messages through its standard input and output channels (clean and simple interface). The DIS (or SIMNET) communications module is the gateway between the local ARSI network and the DIS/SIMNET network. It filters and translates messages going both ways. Other special communication channels can be added to ARSI, such as shared memory pools between software modules, bus-to-bus links, and reflective memory. However, special channels restrict reconfigurability and are usually more complicated than the existing communications architecture.

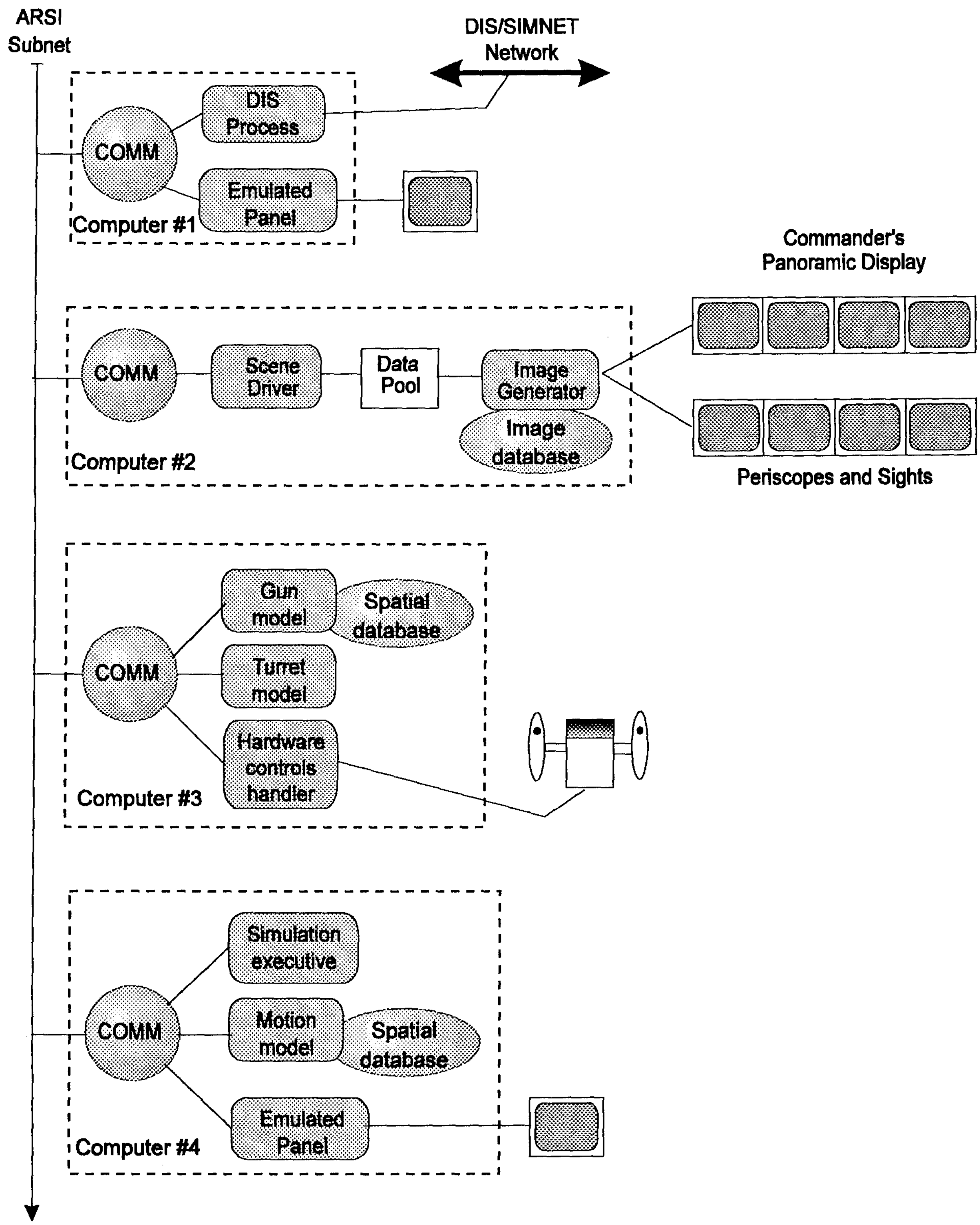


Figure 1. This shows an example of the ARSI software module architecture. The software processes communicate by broadcast message passing through the comm processes. Special communications channels can be used in the modules, such as the shared data pool on the image generator computer. The software modules can be moved between machines (given that any related hardware connections can be moved).

The simulation software modules are not synchronized. Each module is given an update rate parameter, and is responsible for running at the given rate. An unsynchronized system is much more reconfigurable and efficient because it does not use a master timing process to track all the other processes. The configuration must be carefully designed and tested before use to make sure no processes are failing their update rates. Synchronization can be added to specific modules or even the reconfigurable core, but reconfigurability is restricted.

One of the biggest restrictions to reconfigurability in existing simulators is forcing all models to use the database on the image generator. Consequently, we have separate databases for the models and the image generator. Each model has a copy of a query body which runs as a subroutine and answers queries about battlefield data. Separate databases give us the following benefits:

1) The models are free to run on computers other than the typically very expensive image generator.
2) The image generator processes have more CPU time to draw scenes instead of handling model queries.
3) The image database does not have to maintain data that is not necessary for display, eg., trafficability, etc.
4) The spatial database query body has a clean, simple interface that different applications can use. New queries and models can be developed without affecting the existing functions.
5) The spatial database query body answers spatial queries quickly with ANSI C code.

We guarantee correlation between the databases by using the same polygon and model set for producing both the image database and spatial database. This is discussed below. Using separate databases has the disadvantage of using extra computer RAM for storing multiple copies of the battlefield data. This disadvantage is not a problem in ARSI where we have multiple computers to distribute the load. This becomes a problem if a number of models are placed on one host. The query body would have to be elevated to a process that the other models could access.

The ARSI enclosure can hold the whole hardware assembly, including the computers. It is designed to operate indoors. The enclosure has its own air-conditioning. An erector set of floor, frame, and joint pieces brace the assembly. These pieces support the crewstation equipment and support equipment: seats, hardware grips, gunsight eyepieces, cockpit displays, air conditioners, sound equipment, etc. The whole enclosure is covered by canvas panels. Figure 2 shows a picture of the hardware assembly for the M2 configuration.

Figure 2. This is a photograph of the ARSI hardware assembly for the M2 configuration. The whole assembly, including computer equipment and air conditioning, fits in the enclosure. We have rolled up a few of the enclosure fabric panels so the inside is visible.

Configuration state

When ARSI is in the configuration state, it provides a path for defining a new ARSI database based on the S1000 system. We had various reasons for choosing the S1000 toolset:

1) ARSI will be able to game against SIMNET nodes in their existing databases. This was a customer requirement.
2) The spatial database query body and the image generator software use the same set of polygons and models from the S1000 database. This guarantees database consistency between ARSI modules.
3) We do not want to develop yet another database generation toolset and database format. We want to use existing applicable standards.

There are three ARSI routines that leverage the S1000 toolset. Figure 3 shows the tools and the generation path.

For defining a new vehicle configuration, ARSI provides a number of well-documented hardware and software hooks. Here are a few:

1) the erector set of floor, frame, and joint pieces. These pieces can be rearranged into a very wide variety of enclosures The simulator can be broken into separate sections, such as putting the computers in a computer room, or separating the crewstations. Modifying the enclosure typically requires mechanical design time.

2) a configuration file that defines what software modules run on the computers and the parameters for each module. Its format is easily read and edited. Any modules from the library may be included in a simulation.
3) standard communications functions that third parties can use to build new software modules.
4) a configuration file defining the out-the-window visuals. This file sets the number of channels, the fields-of-view and fields-of-regard, sensor type, and a minimum scene update rate. The baseline ARSI prototype is limited to the eight channels on a Silicon Graphics Reality Engine II with two graphics pipelines and multi-channel options. This file is also easily read and edited.
5) the VAPS commercial product from Virtual Prototypes, Inc., for modifying or creating soft controls emulations. This includes the C Code Generator option so that the panels can be executed during an exercise with more efficient code and without the VAPS runtime environment.
6) some table/parameter driven modules. One example is a ballistics model which uses firing table data to model a variety of guns and ammunition types. Another is the hardware controls handler process which has a list of parameters defining how to interpret the analog and digital signals from a crewmember's grip.

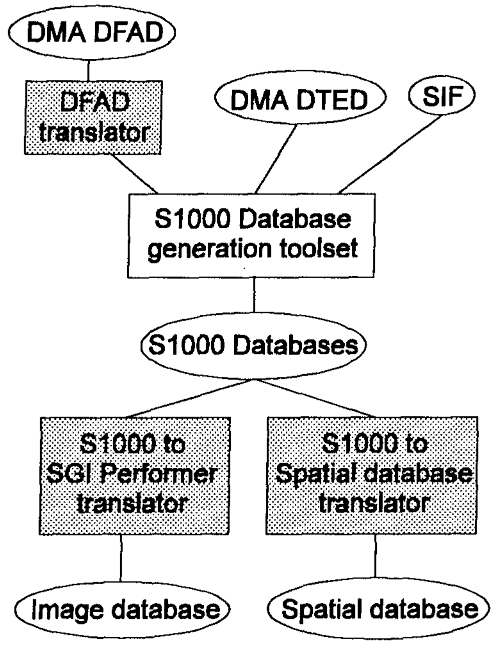


Figure 3. This shows the generation path for the ARSI database. ARSI adds a few tools (shaded) to the S1000 system. Note that the spatial database, which the models use, and the image database use the same polygon set. This guarantees database correlation between the models and visuals.

WHAT IS ARSI'S STATUS?

At the time of writing this paper, we have begun integration and test on the M2 configuration. All five required vehicle configurations will be completed December 1994. The baseline ARSI program is finished in February 1995. Several contract options have been exercised to build eight ARSI footprints (two platoons), design ARSI for a motion platform, and add a setup/logger module.

A number of other changes and additions to ARSI have been discussed. More ground vehicle configurations and aircraft configurations may be added. ARSI is an ARPA-sponsored program directed by MICOM, contract no. DAAH01-93-C-R168.

INSTRUCTOR OPERATOR SYSTEMS: EFFECTIVE DESIGN TO MAXIMIZE STUDENT LEARNING

Linda J. Brent and John B. Heisler
Loral Defense Systems-Akron
Akron, Ohio

ABSTRACT

This paper describes a study conducted during the design phase of a weapons system trainer (WST) for the U.S. Air Force Special Operations Aircrew Training System. The purpose of the study was to identify key instructor requirements of the instructor operating station (IOS) for the WST. During the pre-design and early design phases, an analysis of existing IOS stations was conducted to determine their strengths and weaknesses. The analysis considered instructor tasking requirements, along with task saturation points in the mission training from the perspectives of both student crew members and instructors. The results indicated that many required human factors and instructional design features were not effectively built into many of the existing stations. Several factors also complicated the IOS for this system. The requirement for instructors to have both over-the-shoulder and IOS access to students, combined with the multiple crew positions involved, created complex design problems to solve. Following the analysis of existing systems, a study was developed to determine the critical elements of instructor interface both to the IOS and to the student during both crew station (individual) and weapons system (crew) training exercises. Mission scenarios were designed for use in this study which paralleled real-world situations. The segments of the missions most subject to task saturation for instructors and students were identified. The scenarios were then run under controlled, simulated conditions. The scenarios were videotaped for analysis and systematic debriefing sessions were held following each scenario. The data was analyzed by instructor and mission tasking requirements. Study results were used to define specific design requirements which would meet the instructional needs of the students and the tasking and operational requirements of the instructor. Refinements to the design of the instructor operating stations were made to maximize both the station's human factor capabilities and the instructor-student interactions. General design guidelines are provided for future research in this area.

ABOUT THE AUTHORS

Linda J. Brent is currently the Training Manager for a major AF program, responsible for the integration of training and design requirements for all media developed. Since joining Loral in 1991, she has provided technical leadership for the conduct of two AF schools, curriculum development, and for training/ engineering interface in the final design and production of the training media. Telephone (216) 796-5913

John B. Heisler joined Loral 1983 as a software programmer/analyst in the simulator engineering organization, and was involved in numerous DoD programs in the area of training and simulation. Since 1990, he has served as the Lead Systems Engineer for the full life cycle development (hardware and software) of the instructor operator station for the AF Special Operations Forces program.

INSTRUCTOR OPERATOR SYSTEMS: EFFECTIVE DESIGN TO MAXIMIZE STUDENT LEARNING

Linda J. Brent and John B. Heisler
Loral Defense Systems-Akron
Akron, Ohio

INTRODUCTION

The advance of technological enhancements and procedures maximizes the use of systems engineering, human factors, and training considerations in the design of training media. The use of systematic processes for the integrated design of training systems, to include simulators, has proven difficult both for the organizations completing the design and for the end users, who create evaluation mechanisms to measure the simulator's resemblance to the actual equipment.

In the past, training devices were often provided to the user with the main focus of design on the operation of the device, rather than on the usability of its components, such as the instructor operator station (IOS). In recent years, however, the capabilities of training devices such as weapons systems trainers (WSTs) has expanded dramatically, creating opportunities for crew members to train operations which are difficult to perform in the actual aircraft except in a conflict situation. The rising cost of aircraft use for training purposes has also refocused the attention on use of devices such as WSTs to meet the majority of training requirements. Simulation technology has advanced to the point of creating real-world exercises for crew members, along with the capability for networking trainers to provide further opportunities in full-mission, multiple-aircraft training.

Research and development efforts in IOS design, however, have not followed the pace set by simulation technology (Charles, 1982). While organizations have undertaken research in recent years to evaluate the effectiveness of IOS design features (Charles, 1988), the integration of these features with instructional design and instructor requirements requires further development.

The USAF Special Operations Aircrew Training System includes WSTs for the MC-130E and MC-130H aircraft. Design of these WSTs included several unique features. First, the system was designed to operate in a crew station trainer (CST) mode as a part task trainer for one or two crew positions, as well as in a WST mode for total crew member interaction. The number of crew members in training on the motion-based platform for the MC-130H aircraft totalled five (four crew positions), with a requirement for over-the-shoulder access by the four instructors during training. In addition, the nature of the mission performed by crew members presented unique situations for lighting, position, visual display requirements, and equipment use and availability.

This design, based on the requirements, created a first for industry in the number of personnel required on the platform. The platform designers had the requirements to:

1. Closely resemble the actual aircraft in position, equipment, and so forth.

2. Provide for multiple crew and instructors in a non-interference basis.

3. Provide for conditions in which mission operation can be conducted in a night vision goggle (NVG) environment.

4. Provide ease of access and information utilization for the instructor at the IOS.

These combined requirements created a complex set of considerations in designing both the platform arrangement and the information accessibility of the IOS.

As defined in a study conducted by Braby et al. (1988), IOS station features include the following general categories: training strategy, instructor location, trainer features, instructor features, managing features, and record keeping. The study attended to features in each of these categories. Loral's design of the IOS included consideration of the same general features. In working group sessions held with government and contractor personnel early in the design phase, the applicable features were defined in greater detail. Through this process, the requirements for the IOS station and its use in training were defined, and included the location of instructors (which moved from off-platform to on-platform to provide over-the-shoulder capability); the number of data screens required for instructor/student information; the information needed for the repeater screens located at the top of the IOS; and the means by which instructors would access information during training.

As the requirements and design specifications for the IOS station evolved, it was determined that a study was in order to more clearly define the requirements and to test the accessibility of information by the instructor, particularly during high tasking periods of a training mission scenario. The issues that were explored through the study were not so much about the format of information on the display screens, but rather the broader issues of the overall usability and operability of the system within the training situation. This included: the definitization of the instructor station locations on the platform; instructor accessibility to the IOS station during training; the speed of access to required IOS information during the training scenario; and the capability for multiple-instructor access to IOS information during the training session.

Purpose of the Study

The purpose of this study was to definitize and finalize the instructor operator requirements for the MC-130E and MC-130H WST and CST. The MC-130H simulator, for example, is unique in the number of crew positions and instructors (totalling 9) required on the motion platform (in the cockpit) of the simulator. This factor, along with the complexity of the missions performed and the conditions of mission employment, led to the conduct of this study.

DESCRIPTION OF THE STUDY

Goals and Objectives

The IOS for this program was required to service a large number of instructors in a relatively small area. The space available in the WST was limited by the footprint as well as the height. To address this problem, a team approach was pursued from the inception of the program. The training and engineering organizations, along with human factors specialists, worked closely to develop an IOS that would provide the most "power" for the space allocated.

The team worked together from the requirements analysis phase through preliminary design, holding meetings and discussions with the instructors and training subject matter experts (SMEs) to discuss training objectives and to determine the detailed requirements for the IOS. Early prototypes of the man-machine interfaces and display layouts were used to ensure that the proper technology and design were being pursued.

During the detailed design phase, it was determined that a formal, hands-on study should be conducted. Goals of the study were to familiarize all members of the team with the current design, to identify any deficiencies, and to gather suggestions for possible improvements/enhancements to the system. Up to this point, the hardware configuration had been defined and the majority of the layouts of the display screens had been prototyped. The actual functionality of each had not yet been finalized, but the instructors could view each and determine in a "real-life" situation the most critical elements.

Design of the Study

Throughout the development life cycle of the IOS, standard reviews were held (system walkthroughs, requirements walkthrough, detailed design walkthrough, preliminary design review [PDR]), along with supplemental reviews with the users of the IOS on an as-needed basis (e.g., to address the use of touchscreens versus toggle switch control panels). However, the development team identified the need to perform a more thorough and comprehensive analysis of the IOS and WST as a whole system. To this end, a formal study was organized and conducted.

A room in the IOS design lab was modified to closely resemble the layout of the WST. A mockup was set up to replicate the size and shape of the actual WST. Tracks were located on the floor to control the movements of the instructor seats like in the simulator. The IOS mockup included two screens, two repeater monitors and a work table which incorporated the trackball and other input devices. In addition, the instructors provided foamboard replicas of the cockpit and those were placed appropriately around the room. Video equipment was situated in strategic locations in the lab to tape all simulated mission exercises.

Procedures

A group meeting was held over a three-day period at the facility where the IOS was being developed and built. The meeting was attended by members of program management, the training organization, the engineering development team, WST instructors, and human factors experts. Presentations were made of the capabilities of the WST and the IOS along with hands-on, simulated simulator sessions.

The meeting was opened with a presentation of the layout of the designed simulator station. The group then moved to the mockup, where a brief description of the WST was provided. The IOS design engineers conducted a "walkthrough" of the current IOS screens, provided instruction on how to manipulate the screens, and described the capabilities of the trackball and touchscreens, and so forth.

The simulated WST mission scenarios were presented twice. The first presentation was conducted by the instructors performing the mission as crew members would conduct an actual mission in an aircraft. The instructors sat in crew member chairs, and flew the mission as it would be flown. Instructors used an actual scenario designed by courseware developers for use in the final curriculum. The purpose of this first exercise was twofold. First, it provided the observers with a sense of the entire mission itself, from the student or crew member's point of view. The exercise provided a basis for the instructor simulation which was to be conducted the following day (using the same mission scenario). It also provided the observers an opportunity to view the critical crew coordination and timing issues associated with flying a particular mission, as well as the high task periods during the mission.

Following this exercise, the entire group of observers and participants discussed their observations, problems, concerns, and so forth. The instructors then "flew" the mission a second time, this time in their role as instructors. This exercise demonstrated the ways in which they would use the IOS, determined critical tasking requirements for use of the IOS, and provided visibility to potential limitations.

Two CST mission scenarios were also simulated for the observers. The first scenario was designed for the pilot, copilot, and flight engineer; and the second was designed for the left navigator, right navigator team (as for the MC-130E CST design). These scenarios were actual training scenarios obtained from courseware developers for use in the final training program.

The scenarios were generally run in a real-time mode as they would in a normal training environment. However, interruptions were permitted as required to capture critical data. The instructors interjected comments and recommendations as the scenarios unfolded and the observers asked questions as required. This approach resulted in a good mix of realistic training time and question and answer periods throughout the scenarios.

Description of Scenarios. Three training lesson scenarios were instructed as a part of the study, one full WST and two CST scenarios. They were chosen as typical scenarios for the overall curriculum.

The WST scenario began with the ownship on the ground, the crew having just entered the aircraft. Each crew member was required to go through all checklists in preparation for takeoff. This included all calls and all switch settings. Once in the air, all crew members performed their after-takeoff checklists and activities. Throughout the various phases of the session, crew member checklists were constantly monitored. The scenario then called for a period of low-level flight. Throughout this period, threat detection and avoidance was verified. All crew members were active in the threat detection and avoidance phases. Malfunctions were introduced at this point to simulate the attack by the threats. The crew was monitored for their reactions to these attacks and for their continued efforts to avoid further damage from the threats. Once past the threat area, the scenario called for an airdrop to be performed. After following four more waypoints, the scenario ended with an infiltration/ exfiltration exercise. The crew was required to locate the landing zone (LZ), land there, perform the off/on load procedures and then depart.

The first CST session involved three students and two instructors. It reflected the actions of the pilot, copilot and flight engineer performing an infiltration/exfiltration mission with night vision goggles (NVGs) and was monitored by the instructor pilot and instructor flight engineer. The basic scenario was started from a position on final approach approximately 15 nautical miles (nm) from the landing area and continued through roll-out to a stop, simulated rapid off/on load, 180-degree taxi turn and takeoff, and ended with a clean aircraft at 1000 feet above mean sea level (msl) approximately 2 nm off departure end.

The second CST session involved two students and an instructor. It focused on the actions of two student navigators performing the basics of terrain following/terrain avoidance (TF/TA) flight and was monitored by the instructor navigator. The scenario included positioning the aircraft relative to the selected TF/TA targets, waypoints and radar update targets for the purpose of demonstrating TF/TA procedures and techniques, turn point procedures and techniques, and mission computer updating procedures and techniques, including observation of weather and altitude. Application of these procedures and techniques were monitored throughout the flight path.

Use/Advantages of Note Taking/Videotaping. Accurate recording of the activities, comments, and results of the scenarios was a critical aspect of the study. This was accomplished through note taking throughout the three days of activities as well as through videotaping of the scenarios. The note taking resulted in the compilation of data that was sufficient for publication. The videotapes were analyzed for instructor and student requirements, and archived for future reference in the detailed design process. The videotapes have proved to be invaluable during the continued detail design and production phases and have been used frequently by design and production engineers and training personnel.

During the study, individuals from each of the disciplines (engineering, training, and human factors) took notes. These notes were collected and condensed to create a comprehensive narrative of the activities as well as to obtain a thorough list of the observations and recommendations.

The scenario sessions were videotaped from the key viewpoint, concentrating on the IOS and the instructor pilot, who was the chief narrator of the scenarios. Though the actual information on the IOS displays was unreadable, it provided good insight into the number of times the instructors required access to the information on the displays, the number of times they needed to interact with the displays and their accessibility to the IOS in general. The audio portion of the videotape was also very important. It provided the actual communications among the instructors and crew members concerning performance of their tasks as well as comments on the current design; and comments of the observers who were present.

LESSONS LEARNED

Summary of Results

The presentation and discussion sessions, as well as the mission scenario sessions, provided valuable results. The discussions were lively in both sessions and a number of recommendations, suggestions, and design confirmations were generated.

Relative to the overall layout of the platform, it was determined through these exercises that the required number of instructor and student/crew personnel could be accommodated safely on the motion platform. Egress from the front seats of the cockpit was discussed, and a suggestion was made to install a folding seat for the flight engineer (FE) instructor, which enabled easier and safer egress from the front cockpit seats. The location of the FE instructor was also discussed in terms of access to the circuit breaker panels. It was concluded that the FE instructor could act on verbal command of the student(s) for any activity required at the panels obscured from the FE instructor. Another issue related to the station layout dealt with the NVG curtain (designed and located as in the aircraft). The results of the study concluded that the curtain need not be closed entirely at any point since the lighting at the IOS is all NVG-compatible. It was felt that the curtain would be used across the right-hand portion of the flight station to cut down on the glare from the lights at the navigation/electronic warfare operator (NAV/EWO) crew stations panel(s).

The next issue discussed concerned the instructor techniques when using the IOS. The issue of over-the-shoulder versus remote instruction was highlighted as a critical factor in the ultimate success of the IOS. The outcome of this exercise as to the need for both over-the-shoulder and IOS station access varied from instructor to instructor. The results indicated that the instructor viewpoints varied and were dependent upon past simulator experience. Those instructors who had little or no simulator experience were most comfortable with the over-the-shoulder technique that they were most familiar with; those instructors with prior simulator experience were able to provide more substantive comments concerning the appropriate utilization of the IOS station and the information provided. The participants familiar with instruction using simulators stressed that use of the IOS in its current configuration would be extremely beneficial in their training activities.

The results of the study relative to the CST mode of operation were discussed at length. The design of the CST mode provides for the training of multiple students at the two stations simultaneously. For example, while one student pilot and instructor are training on one scenario at one of the two stations, the electronic warfare student and instructor can train an independent scenario at the second station. The requirement for simultaneous training resulted from the expected student throughput for the training sessions in the schoolhouse. Much discussion centered around the results concerning the requirements for functionality and fidelity while in the CST mode of operation. It was determined that further analysis of the requirements was warranted to ensure that the engineering design met all training and throughput requirements, based on the curriculum design.

The Electronic Combat Environment (ECE) makeup was also discussed as it relates to the training environment. Initial requirements for the WST and CST were defined for use in mission rehearsal. It was determined that the requirements for ECE training were not as complex in either the WST or CST training, since the design is common across both the WST and Mission Rehearsal Devices (MRDs). Further discussion and study in this area was not warranted at the time.

The group next focused on the results found during the initialization process of the WST/CST. The training organization and instructors expressed the need to automate as much of the mission scenario exercises as feasible. It was determined that the planning software configuration must include ownship location, ownship flight path/plan, ownship configuration (fuel, oil, armaments, etc), as well as malfunctions, threat laydown and actions, and weather conditions. It was recommended that this information be organized into sets which would then be grouped logically as missions. It was also determined that it would be desirable to have the capability to use varying sets of information within a mission scenario in order to individualize training to meet the training needs of all students/crews. Selection of a pre-defined mission should be straightforward and should be viewed as the standard mode of operation for training exercises.

Results of the study were also discussed concerning the repeater displays, the information required, and their functionality. The existing design allowed the repeaters to display (repeat) the video simultaneously displayed at a selected student station. It was agreed through the discussions and study analysis that out-the-window views and video views available to the student crew would not be required, since this information is available in the over-the-shoulder view of all instructors.

These and additional issues identified during the WST scenario study can be summarized as follows:

1. The arrangement of instructors and student/crew on the platform was workable, with some modifications for the FE instructor.

2. The number of IOS screen utilization conflicts between instructors was less than anticipated. Results indicated that the instructors shared utilization effectively, even during the high tasking periods of the mission scenarios.

3. The FE instructor, in his position on the platform, was totally removed from participation in the scenario through his use of the IOS. He was unable to see or reach the IOS.

4. The remote hand-held device used by the FE instructor, as designed, was rarely used. It was determined, however, that a hand-held device with additional capability to access instructor information and input data would be used by the FE instructor and would make him a more active participant in the training scenario. It was determined that the ability to have a display

capability on his remote is essential to bring him into the training situation.

5. The display of a background chart (map) on the IOS instructor screens would be a great enhancement for navigator and EWO instructors.

6. Overall, the format and information arrangement on the display pages were found to be quite effective for use by the instructors. Some pages were reorganized for more efficient utilization, including the status, approach, departure, rendezvous, and communication pages.

The study also identified several additional potential design modification requirements. These included:

1. The content and arrangement of the status information screens required some revision. The instructors experienced some utilization conflicts in screen access of status information.

2. The FE instructor, as stated previously, was functionally left out of the scenario due to the lack of free access to the IOS due to his physical location on the platform. He was unable to clearly view the IOS screens from his position. It was determined that an enhanced hand-held remote would allow him to access information from the IOS.

3. It was determined that the ability to monitor the communications transmissions by the students is critical. Thus, it was recommended that the communications capabilities accessed through the IOS be modified.

4. Human factors personnel identified the following additional observations:

    a. The pilot instructor must twist his body backwards to use the IOS. It was determined that this was not a major problem.
    b. The trackball position on the IOS shelf did not permit access by all instructors at the station at the same time. Modifications to the location of the trackball were recommended.
    c. The ability of the instructors to move the seat positions quickly was hampered by the crowding on the platform. It was determined that this could not be completely assessed at the time of the study due to the lack of actual seat equipment used in the scenario.
    d. The EWO instructor had difficulty seeing the forward multi-function displays (MFDs) in the cockpit. Likewise, it was difficult for the pilot instructor to view the aft IOS screen. These items were determined to be of no major consequence, as the required information could be accessed by the instructors on a nearby IOS screen.

5. The instructors required an out-the-window view, although it was determined that this would be available to them from their position on the platform.

The following issues were identified specifically during the CST sessions:

1. The navigator instructor's manual control of the "dummy" aircraft to follow verbal heading changes was discussed and determined to be usable in its present design configuration.

2. It was determined that the pilot instructor should be able to insert radar updates into the IOS.

3. It was requested that engineering reconsider the ability to have a record/replay capability in the CST mode of operation.

Implications/Applications of Results to Current Design

The issues noted during the conduct of the study resulted in a series of general recommendations. The following represent key recommendations from the study and their disposition:

1. The ability to perform record/replay while in the CST mode is a requirement for training, and will be provided on one station at a time. For example, if the pilot/FE station is conducting record/replay, it will not be simultaneously available on the NAV/EWO station. It was determined by all to be sufficient for training requirements, and capable of easy integration into the device.

2. The current design of the hand-held device has some limitations which were identified by the study. As a result, its capabilities were reevaluated and a revised design was implemented

which meets the training requirements. This also solved the problem of the location of the FE instructor -- he was given additional capability on the hand-held device to meet his requirements.

3. The instructor EWO must move back from the student to the IOS to manipulate or modify the threat. This problem was addressed in the redesign of the hand-held device, giving him additional capability which permits him to stay in the over-the-shoulder position with the student.

4. The instructor is required to manually control the *dummy* aircraft during the CST mode of operation, to include radar updates. This change was incorporated into the final design of the IOS.

5. The need for additional screens and more windowing capability was identified. This was added to the final design considerations of the device.

6. The need for the instructor to determine actual aircraft position, mission computer position, and planned position at the IOS was identified and provided in the final detailed design.

7. The capability for the instructor to monitor the cumulative effect of battle damage to the aircraft was identified and provided in the final design. Note: The IOS also provides the capability for the instructor to set a cascading malfunction scenario, based on the probability of survival data maintained by the simulator.

8. The capability for a map background chart on the IOS was determined to be proposed as a future upgrade to the system.

Overall, results of the study indicated a user-friendly IOS station, with ease of information access and the capability for multiple status display information through windows and function keys. Instructor-student interaction was effective with both the over-the-shoulder capability and instructor use of the IOS. It was further determined that the design and utilization of space on the platform was excellent, providing ease of access for instructors along with aircraft design characteristics for student crews.

FUTURE DIRECTIONS

Analysis of Study

Positive Outcomes. The development team agreed unanimously that the study provided useful and concrete design feedback. Having the instructors and training personnel in a situation in which they worked side by side with the engineering development team provided insight to both organizations of what was required to meet the training requirements. All too often, the integration of these groups is not sufficient in the design of the devices, and this can lead to devices which fall short of expected or required capabilities. In an aircrew training system (ATS) program, it is critical that the engineering and training organizations work together closely to meet all requirements. The entire team must be involved to ensure that the student, instructor, and media come together with all requirements met in a way to maximize the training opportunity given a variety of training scenarios. The study proved useful for the engineering development of the system, the training/instructor development of the curriculum, and the human factors elements of the IOS.

In addition to providing instantaneous feedback for design considerations, the study also created a situation which opened communications between these organizations for the remainder of the program. Often in programs of this type, the communications between engineering, training, and end users is not as it should be for effective, efficient design and use of the device. This study forced these organizations to come together, test assumptions in a controlled environment, and analyze the results systematically, leading to increased understanding among all concerned.

Videotaping the study's mission scenario exercises also proved to be extremely useful. The actions and activities of the instructors were captured for ongoing analysis. The audio portion captured the dialogue throughout the exercises, and served as a permanent record for reference during final design. These tapes were put to immediate use following the conduct of the study by members of the development engineering team who were not present at the sessions. Follow-up reviews of these tapes yielded additional observations and viewpoints.

Pitfalls to Avoid in Future Studies. As is often the case in the design of devices, past experience on the part of the users and personal preference can influence the design outcome. Although this was also the case in this situation, personal preference was often overcome by the objective conduct and outcome of the scenario procedures. Thus, the study created opportunities for decisions to be made based on objective observations rather than impressions of instructors and training personnel. The analysis of the study also included the identified personal preference recommendations. These were then weighed against the results demonstrated through the taped sessions.

A second factor in this study was the size of the lab room. While it closely resembled the size and configuration of the WST, it limited the ease of observation by others and the flexibility to position the videotape equipment. It did, however, create a realistic environment in which to conduct the study. One solution could be to provide a larger space with the footprint of the device marked in the space as appropriate.

Guidelines for Future Studies. A number of guidelines can be generated from the conduct of this study which can be useful in future endeavors of this type.

1. The timing of such a study is critical. While it should be done early in the design cycle, it is also important to have the initial design parameters completed in order to make the session productive. Depending on the size and complexity of the device, it may also be useful to conduct such an event at several intervals during the design and development process. If multiple session are conducted, however, a core team of individuals should be involved at all stages to ensure consistency of design and history of development.

2. To minimize the personal preference inputs by users and designers, a well constructed set of scenarios, consistent with the training requirements, should be employed. Each resulting design consideration evolving from the study should be analyzed in an objective manner, and should be data based.

3. It is most effective to involve a team comprised of those with training expertise, engineering expertise, as well as the end users from both an instructor and student point of view throughout the study. Personnel experienced in the aircraft in addition to experienced instructor personnel can provide insight unequalled in creating a realistic test scenario.

4. Creating an environment which realistically depicts the eventual training situation provides for more believable results. Participants, if comfortable in the environment, can role play the simulated mission scenarios more effectively, and the results are much more credible. This provides more accurate representations on which to make design considerations.

5. Participants should accurately represent the anticipated users of the final product. If the skill and experience levels of the study participants reflect the end users, the data generated is a more accurate representation of training needs and requirements. Additionally, those who analyze the results should reflect all disciplines, (e.g., engineering, training, and human factors). All perspectives are required to make effective design decisions.

SUMMARY

The results of this study provided an objective basis upon which to make final design decisions for the MC-130E and MC-130H WSTs. The conduct of the study brought together all disciplines, working as a team, to ensure that the final training requirements were met. Although research has been conducted on an ongoing basis concerning the most effective, efficient operational features of IOS systems, the study described here provides two additional design considerations. First, the conduct of the study brought together engineering, training, and users to collectively define the design to meet the unique characteristics of this device. All viewpoints were considered prior to final decisions, and agreement was reached by all team members. In such a way, the considerations of all perspectives were made and decisions were data-based, and interpreted from all perspectives.

Additionally, design decisions were focused on the requirements of the end users (student crews and the instructors) rather than on ease of design from an engineering perspective or a human factors requirement only. Designers of media devices must, as technological capability increases, emphasize the requirements from the student and user perspective. Otherwise, designs can be driven by technology capabilities alone, creating circumstances in which a device is over-designed and unnecessarily costly from a training perspective.

REFERENCES

Braby, R., Charles, J., Sylla, C., Ramesh, R., Willis, R., & Hunter, D. (1988). Instructor considerations in the design of optimal training devices. (Report No. AD-A252 122). Orlando Field Unit, Orlando, FL: U.S. Army Research Institute.

Charles, J. P. (1982, August). Operational problems in instructor operator station designs. Paper presented at the Workshop on Instructional Features and Instructor Operator Station Design for Training Systems at the Naval Training Equipment Center, Orlando, FL.

Charles, J. P. (1988). Instructor/operator station (IOS) design guide. (Report No. AFHRL-TR-87-32, AD-A192 055). Brooks AFB, TX: Air Force Systems Command.

THREAT SIMULATION: TRADEOFFS BETWEEN TACTICAL REALISM AND TRAINING VALUE

Samuel F. Bass
AAI Corporation
Hunt Valley, MD

ABSTRACT

Threat simulation in electronic warfare training requires both signal fidelity and tactical realism. These aspects of simulation are generally not in conflict. However, as tactical realism is increased -- through the use of autonomous tactics models responsive to simulated ownship position and crew countermeasures -- training value can be compromised. Specific problems can include: inability to "schedule" the hostile signal environment to avoid trainee overload or to present very specific signal combinations; loss of insight into exactly what situation confronted the trainee at any given moment; and loss of repeatability in a given mission, hence loss of the ability to deliver equivalent, objective-oriented training to successive trainees.

Modern training systems must balance these issues to assure the development and maintenance of superior skills in the electronic combat community. This paper describes the tradeoffs to be considered in the design of threat libraries, selection algorithms, and tactics models. It further indicates approaches to be considered as a function of purpose of the simulation and the level of training to be delivered.

About the author

Mr. Bass is the lead human factors specialist at AAI Corporation, Box 126, Hunt Valley, MD 21030 (410-628-3902). He has been responsible for the definition of instructional, performance measurement, and user/computer interaction features for surface/subsurface ASW and electronic combat trainers and flight simulators.

THREAT SIMULATION: TRADEOFFS BETWEEN TACTICAL REALISM AND TRAINING VALUE

Samuel F. Bass
AAI Corporation
Hunt Valley, MD

INTRODUCTION

Threat simulation in electronic warfare training requires both signal fidelity and tactical realism. These aspects of simulation are generally not in conflict. However, as tactical realism is increased -- through the use of autonomous tactics models responsive to simulated ownship position and crew countermeasures -- training value can be compromised. This paper will examine several potential sources of training effectiveness degradation and present strategies for avoiding their detrimental effects.

REQUIREMENTS FOR COMPETENCY IN ELECTRONIC COMBAT

Competency in electronic combat requires more than the relatively straightforward ability to read computer-generated presentations of threat activity. It demands excellent cognitive and associative skills, which the specialist must apply in a time-critical, often high stress environment. Training of these higher level cognitive skills requires high fidelity, highly interactive simulations. These high fidelity simulations are mandated by the following factors.

- The enemy does not cooperate. There is no peacetime environment or training range which presents sufficiently dense and interactive hostile conditions to allow training in real-world conditions.

- There is no way to reliably train or test personnel and their systems utilization skills in the absence of a realistically interactive environment, and have any faith in their ability to handle the workloads and stresses they will meet in the real world.

- Reduced fidelity, part-task or "most-task" training experiences do not train or test electronic combat processes in the context of actual mission performance, with its additional requirements for crew coordination, navigation, communications, safety of flight, or ordnance delivery.

This situation is not peculiar to electronic combat training alone. It applies to all sensor-intensive combat domains, such as surface naval warfare and the training of submarine attack teams. The complexity of high fidelity simulations for these combat domains stems from the need to (1) replicate the relevant portions of the physical environment, whether RF, acoustic, etc.; (2) choreograph both friendly and hostile sensor evolutions; (3) model and automate the sensor environment's complex responses to inputs made by personnel in training; and (4) maintain sufficient control of the training exercise to assure the delivery of a valid training experience with sufficient data collected to provide timely and effective post-training debriefs.

The Three Aspects of Threat Simulation Fidelity

In the context of electronic combat training, simulation fidelity has three aspects: signal fidelity, equipment fidelity, and environment fidelity. These are discussed below.

Signal Fidelity. Faithful signal simulation in terms of parameter fidelity is an absolute requirement if students are to learn to recognize signals by aural and visual analysis. It is also an absolute requirement if the simulator must stimulate operational receiver processors. Relationships among all parameters and all signals must be correct, particularly as an emitter transitions from mode to mode. These transitions, while often difficult to accurately simulate, are important for the cues they offer the electronic warfare specialist.

The supporting software must provide convenient access to the details of signal parameters, so that (1) multiple examples of the same emitter types (not exact copies) can be presented; (2) complex pulse trains (various jitters and staggers, frequency modulation on pulse, coded data pulses followed by CW blasts, etc.) can be tailored; and (3) so that signatures can be kept concurrent with the latest signal intercept data.

- **Equipment Fidelity.** Simulation of control operations and display formats is not enough. Electronic combat specialists must learn to recognize and cope with the aberrant behaviors and deficiencies of their equipment and the signal environment. Some of the effects of interest here are ambiguities and anomalies arising out of receiver processor software; noise that rides on received signals; improper antenna selection; and changes in observed pulse shape and width as a function of receiver saturation, bandwidth changes, or off-tuning. Of course, the specialist cannot be exposed to these effects unless excellent correlation is maintained across both simulated inputs and audio and video presentations on all simulated equipments, such as receivers, DF and omni antenna systems, analysis displays, pulse and spectrum analyzers, etc.

- **Environment Fidelity.** A comprehensive simulation of the electronic combat environment is needed to train higher level electronic combat tasks. These include decision making, maintenance of situational awareness, ability to anticipate, task prioritization, and resource allocation under the stress of combat. To close the training loop, however, the specialist in training must also see the results of his activities. The integrated electronic combat environment must respond to the combination of changes in applied countermeasures, changes in the trainee's platform position or activity, and changes in the position or activity of other simulated entities, such as friendly strike packages.

Thus is mandated the requirement for tactics models for each sensor system in the environment. These tactics models emulate the performance of both the sensor and its human crew. If they are not included in the simulation, the training experience becomes a series of disconnected episodes from which the dynamic nature of real world combat is missing.

TRAINING PITFALLS IN TACTICS MODELING

Tactics models, however, must be implemented with due regard for the limitations of the students who will face them. The following attributes of tactics modeling used for training can lead to degraded training effectiveness.

- Real-world tactics models, as do real world sensor and weapon systems, do not forgive student errors.

- Real-world models do not support specific training objectives.

- Real-world models lead to rapid changes in the student's instantaneous combat situation.

- The randomness of real-world models can destroy the repeatability of training.

Each of these pitfalls is discussed below.

The Unforgiving Nature of the Models

Real-world weapon systems -- and their operators -- are expected to inexorably push for victory and exploit every weakness in their adversaries. However, it is not appropriate to subject students to such models in the skill acquisition and early skill demonstration phases of their training. To do so leads inexorably to student overload, as each misstep provides the simulated environment with incremental advantage. In short order, the environment has "ganged up" on the student, his training session is beyond his control or comprehension, and he has failed. He is demoralized, and he has learned virtually nothing.

Failure to Achieve Specific Training Objectives

Tactics models generally include built-in elements of randomness, so that a threat does not behave in exactly the same way every time it is encountered. This randomness may affect time spent in a given operating state; whether a specific state is even entered; responses to student countermeasures; probability and nature of terminal engagements; time delays in transmitting targeting data throughout a threat network; and so forth.

The pseudorandom nature of the models is designed to provide "realism" for the student. Unfortunately, the tactics models pay no heed to training objectives. They are not interested -- as training developers and instructors should be -- in presenting specific tactical situations for which desired student behaviors can be defined and observed. In many training applications, the models have wrested control of the specifics of threat mixes, signal activity, and engagements, away from the instructors. The consequence is often hodge-podge training evolutions in which instructor feel for student performance is substituted for objective measurement.

Rapidly Changing Combat Environment
The situation is exacerbated when multiple threats and multiple platforms are operating under autonomous models. The instructor/ evaluator might then work even harder than the student. While the student must maintain situational awareness, analyze the environment, and make equipment utilization decisions, the instructor must do all this, plus note (and quite often, remember) what the student did that was suboptimal or plain incorrect. Thus neither the student nor the instructor can reliably maintain good insight into the details of a complex encounter and how the encounter was handled. When numerous encounters are presented in an extended training session, debrief and remediation are quite often cursory and lacking in the crucial detail which the student needs to understand and correct his behavior.

Loss of Repeatability of Training
The random elements of threat modeling can make it impossible to precisely compare multiple students to a performance standard. No two students are confronted with exactly the same situations, even though both are exposed to the same environment. The student skilled in the early phases of engagements may earn himself a "milk run" and not even be thoroughly tested in the more stressful later stages. The student who is inattentive early in the game will have to work harder to "pass." Neither student can be considered fully trained.

POTENTIALLY POOR OUTCOMES

Typical of the undesired results of these problems are those below.

- The student becomes reactive rather than responsive to the environment, as simulated threats take advantage of his mistakes and press him ever more closely. He may carry excess anxiety from training session to training session, becoming progressively less effective as his training proceeds.

- The student becomes "focus trapped," attending only to those portions of his equipment and the environment which arrest his attention, and resorts to stereotypical behavior when he should be maintaining a disciplined, ongoing analysis of the environment and offering well thought-out responses to his situation.

- Random aspects of modeling are particularly problematic in early training, because specific student actions can elicit an assortment of threat responses. The student cannot reliably identify cause and effect, a situation leading to ineffective reinforcement of lessons learned.

- The instructor cannot reliably connect student action with the specific precipitating event. More important, the instructor has no record of other simultaneous events which may have mediated the student's thought processes.

- The instructor cannot conveniently arrange engagements which will either demonstrate specific points to a student or which will require the student to make known responses at known times.

- It becomes very hard to demonstrate that all students have been trained to a specific performance criterion.

SPECIFIC PALLIATIVE MEASURES

These problems can be avoided or mitigated during scenario development, the actual running of the scenario, or post-scenario debrief, as discussed below.

Scenario Development Measures

To prevent purely numeric overload -- both of the student and certain simulated receiver processors -- provide the capability to limit the number of threats which can populate the environment at any moment. Limiting parameters can include the following:

- maximum number of simultaneously active threats

- maximum number of simultaneously active threats in a given class, e.g., early warning radars, acquisition radars, IFFs, antiaircraft artillery, surface-to-air missile (SAM) fire control radars, etc.

Care must be taken, however, in developing the prioritization scheme used to select from the available entities. For example, selection by range and lethality alone will not work, because the environment can become overloaded with short range SAMs, and all early warning radars will be discarded. A better method might be prioritization based on the order in which the most astute tactician would choose to bring the threats into the problem. All methods have their drawbacks, and customization for the specific training and simulation application is highly recommended.

After a satisfactory method of threat selection is adopted, the problem becomes one of choreographing the moment-to-moment interactions with the student. Several features should be implemented, the first of which is to include the ability to inhibit or delay the initiation of an engagement. Simply specifying a time for activation or inhibiting activation is usually sufficient. A second feature would be to provide for the delay or inhibition of critical phases of an engagement, such as entering a tracking or shooting state. Delaying or prolonging these phases gives the student more time to detect, observe, and respond to these events. Inhibition capabilities can be implemented in any number of ways, such as:

- limiting the number of simultaneous tracking or shooting evolutions;

- limiting the number of such evolutions requiring the same equipment for countering;

- limiting the number of evolutions occurring in the same quadrant around the student's platform;

- inhibiting engagements requiring contradictory responses on the part of the student, such as evasive maneuvers in opposite directions, release vs withholding of expendables, and so forth.

This last item presents a quandary, in that decision making training requires that students be presented with seemingly conflicting situations. The intent is to control these situations such that there exist one or more clearly discernible correct responses and such that each potential student error can be used to deduce the nature of the student's deficiency, e.g., book learning, signal recognition, or misunderstanding of a specific aspect of the tactical scene.

A largely ignored aspect of scenario development -- even if many of the preceding recommendations are followed -- is the implementation of user interfaces that simplify the developer's task. In many cases, the developer must "prefly" the mission, make notes regarding problem areas, rescript, recompile, and refly to see if the problems are alleviated. This is incredibly time consuming. Given increasing customer desires for rapid scenario implementation, future system specifications will not accept such methodologies.

A better level of implementation consists of providing the developer with an interactive graphic preview of the scenario, showing the positional and temporal relationships among all platforms and threats. Positional relationships are usually shown in a PPI-type format on a combat situation display. Temporal data is perhaps best depicted in a timeline format, with the use of color coding and symbology to indicate the status of each entity in the scenario. The developer then inspects the displayed data by requesting the instantaneous combat situation as a function of student platform location or scenario time. He is then provided graphic and tabular data detailing simultaneous threat activity, relative position, and so forth. Given this data, he can then edit (both graphically and in text) the scenario file to create the required training situations.

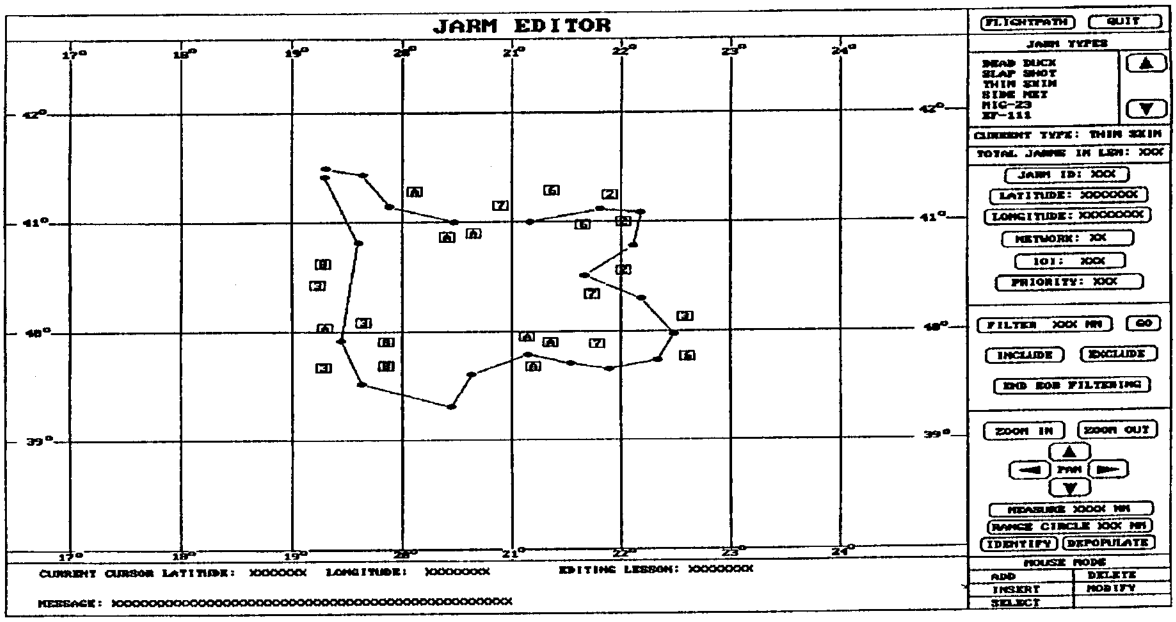


FIGURE 1: Sample PPI Display. Allows selection, positioning, and filtering of threat types, and their position relative to ownship's flight path. Lists and allows editing of all threats simultaneously active at ownship's current position.

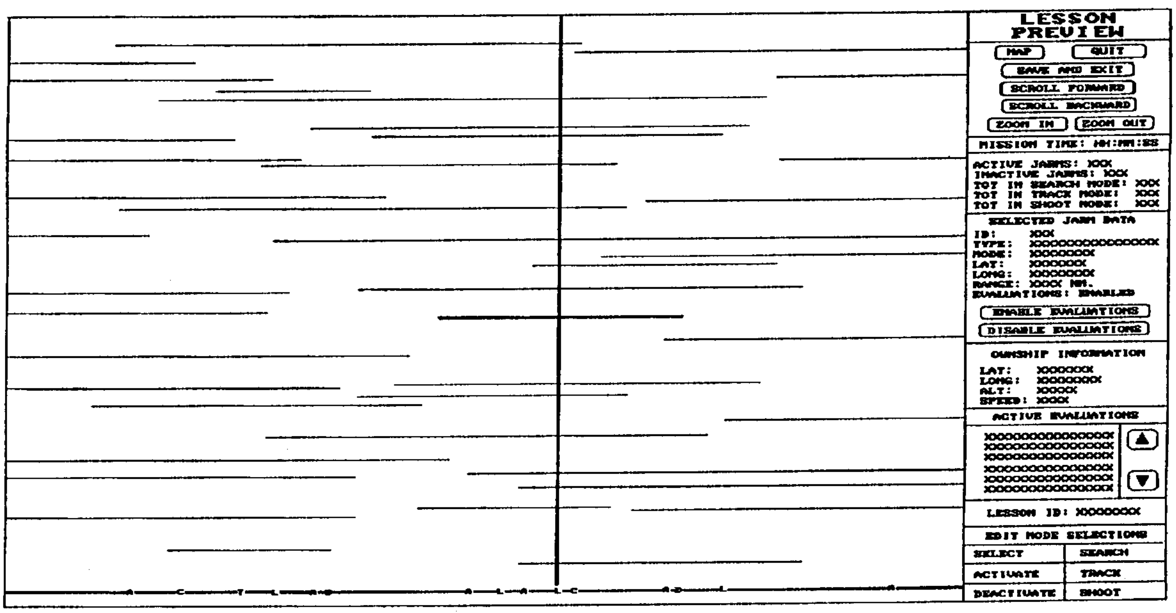


FIGURE 2: Sample timeline display. Shows active time for each threat in relation to all other threats. Allows same editing capabilities as the PPI display. Correlation of time and ownship position is provided by crosshair symbology on the timeline display and highlighting ownship symbology on the PPI display. This allows immediate determination of the student's instantaneous workload.

While this graphic interaction method is a vast improvement over "fly and fix," it is still relatively cumbersome and time consuming, in that the developer has to review the scenario on a moment-by-moment basis, maintaining his own awareness both of the tactical situation and of the training objectives. A more advanced solution consists of implementing "watchdog" software. Such software has the following characteristics:

- it is "taught" the capabilities and capacities of the simulated sensors and countermeasures equipment;

- it is "taught" an assortment of logical relationships regarding acceptable tactics, e.g., an aircraft cannot simultaneously break away from two threats when one is on either side of the aircraft, or a manual jammer may not have its bandwidth spread greater than some number of MHz;

- it monitors the instantaneous tactical environment for signal presence and activity (searching, tracking, shooting, etc.);

- it examines the environment in accordance with rules and queries set up by the scenario or curriculum developer;

- it alerts the developer to undesirable situations based on its own logical rules and special developer-inserted rules and queries.

The developer's rules and queries are important for creating scenarios most appropriate for given levels of training. They might take the following forms:

- alert when more than three SAMs are simultaneously engaging the student;

- alert when 10 direct threats are simultaneously active;

- alert when several threats requiring different modulations must be countered by the same manual jammer;

- alert when all search radars have been pushed out of the environment due to the insertion of higher priority signals;

- alert when three signals all must be countered by the same piece of equipment;

- in conjunction with the threat selection algorithms, **do not permit** some or any of the above situations to develop.

This last capability is invaluable. It greatly reduces scenario development time, because the developer need not search for nor fix these problems. The scenario almost builds itself, and the level of difficulty is appropriate for the student, as dictated by the curriculum and training objectives. No situation is presented which is beyond the (presumed) capabilities of the student or the simulated equipment, yet all possible realism and tactical responsiveness are maintained.

Measures Applied During Mission Run

After the student has nested in the simulator, three approaches are suggested for improving training value: careful limitation of instructor features for control and modification of the scenario; appropriate presentation of integrated environment and student performance data to the instructor; and methods of alerting the instructor to critical problems outside his immediate purview.

Limitations on Instructor Control and Modification Features. Instructors traditionally want the ability to change or control everything in a training scenario. They will add threats, initiate unscheduled attacks, introduce malfunctions, increase the noise added to comm channels, and generally do things intended to keep students on their toes. As a result, criterion referenced training goes out the window. Students are not trained to an objective set of standards, but to the internalized -- and often unverbalized -- standards of the instructor who happens to be on the console that day.

Minimization of these adverse effects requires careful attention (1) to what instructor capabilities are provided, and (2) to the careful integration of those capabilities with the simulator's system for monitoring and recording student performance. The second point is addressed first.

Student Performance Monitoring and Recording. The essence of the problem here is separation of the wheat from the chaff. It does no good to record everything for replay, because instructors rarely will have time to use a replay feature. The simulator is overbooked, and the instructor usually is as well. It is similarly inadvisable to rely completely on the instructor's memory and predilections regarding critical student behaviors. It is possible, however, to collect relevant data on individual engagements, sort them as to degree of student success, and if desired operate statistically on the results. The following are possible data collection categories.

- number of threats responded to within a certain time criterion (measures student situational awareness)

- types of threats responded to within a certain time criterion (differentiates situational awareness from threat recognition and prioritization);

- numbers and types of threats countered using incorrect equipment techniques, e.g., wrong modulations, etc. (highlights either student's lack of understanding of acceptable threat counter, or his lack of facility with his equipment);

- numbers and types of threats misidentified (depending on available simulated equipment, identifies inability to interpret display symbology, inability to correctly measure parameters, or deficient book learning, leading to incorrect association of threat parameters with threat identification);

- in a reconnaissance setting, number and type of parameters logged with incorrect equipment setups (attempting to identify a scan type while receiving the signal through a rotating direction finding antenna, indicating failure to follow proper procedures).

The list of such categories is endless. The point, however, is that such data can be collected. It can be stored for a very efficient post-mission debrief with the student. It can be used to identify required remedial training. And it can be presented in real time to the instructor to direct his attention to problem areas and allow him to counsel the student "on the fly," so to speak. A proper system for alerting the instructor to undesirable levels of student performance further reduces instructor workload and contributes to the efficiency and efficacy of the training session.

Instructor Capabilities for Scenario Modification. In a well-structured curriculum, instructors should be actively discouraged from tampering with carefully constructed training exercises. Nevertheless, it is inevitable that the need for impromptu remedial training, special demonstrations, or the occurrence of unforeseen student ineptitude will dictate that instructors be provided with modification capabilities. The requirement is (1) to limit the capabilities available during formally constructed scenarios, (2) to maintain comprehensible records of changes made, and most important, (3) to be able to interpret the effect of the instructor change on student performance to formally defined criteria.

Potential Application of "Pseudo-adaptive" Training

Adaptive training can be loosely defined as an automatically controlled training evolution in which the problem becomes easier or more difficult as a function of a student's moment-to-moment performance. For the purposes of this paper, pseudo-adaptive training (PAT) is similarly defined, except that adaptations are under instructor control.

Successful implementation of PAT requires that training objectives be defined in a slightly different fashion than is the current custom. Objectives are now stated in the general form of, "....counter all presented threats, within the time criterion appropriate to this level of training, using proper tactics." PAT objectives would require the inclusion of additional "terms" in the equation. For example:

> "At training level "X", counter all presented threats within time criteria and using proper tactics, **where the threat mix and tactical situation shall be defined in terms of level of complexity and tactical competence."**

SEGMENT / BIN NO./TITLE	1	2	3	3R	4	5	6	7	8	9	10	SUM/ COUNT	AVG
1 COMM SET													
1. THRT WARN		18/24	12/24	15/18								30/48	2.8
2. CALL MAN.	--	6/12	3/12	9/12								9/24	2.5
3. EXT. RESP.			8/8	8/8									
4. INT. RESP.	16/16	8/8										32/32	4.0
5. XXXXXXXX													
6.													
7.													
8.													
9.													
10.													
2 JAM SET													
11. TIME ACC													
12. FREQ													
13. BW ACC													
14. MOD ACC													
15.													
16.													
17.													
18.													
19.													
20.													

FIGURE 3: Sample Performance Data. Data can be presented as score sums, averages, or counts of occurrences.

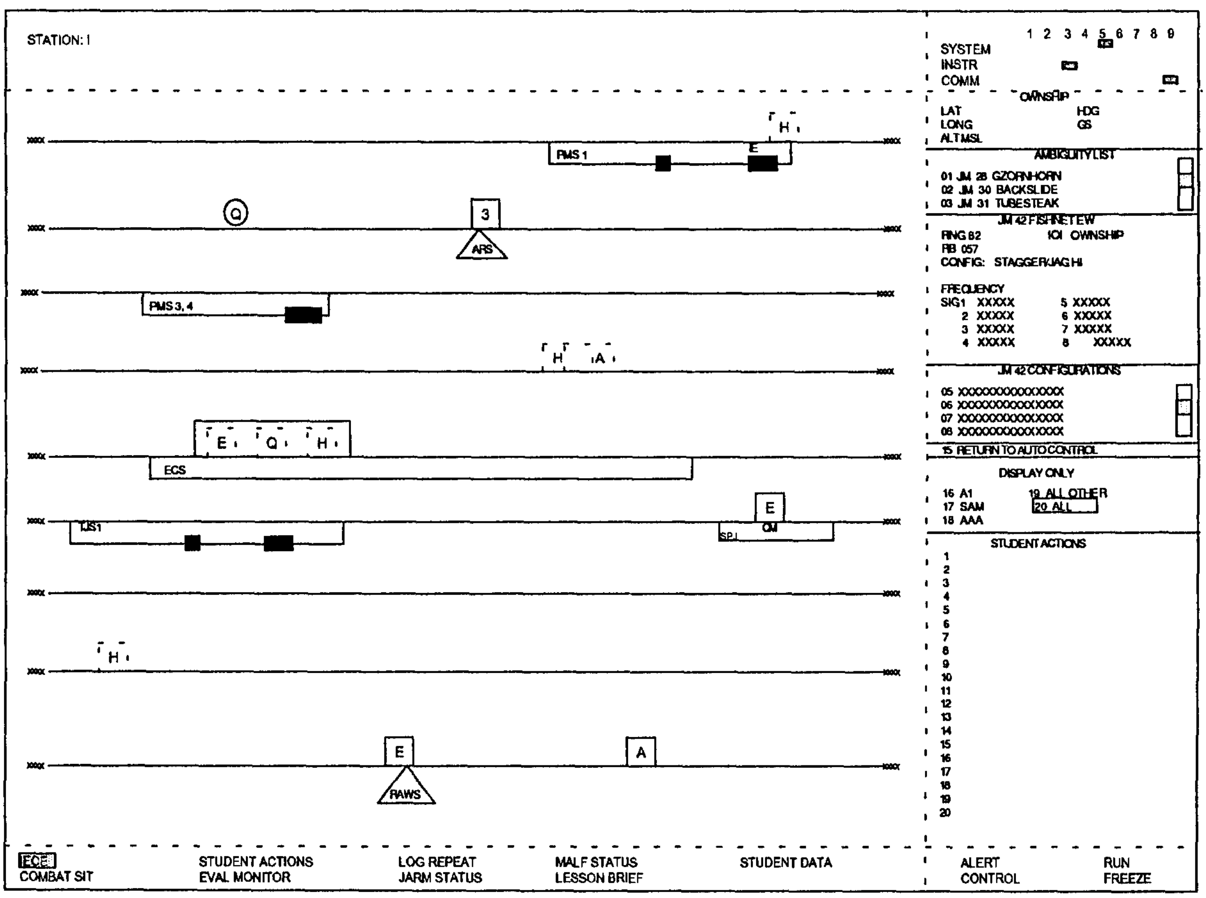


FIGURE 4: Instantaneous Electronic Combat Environment Summary. Data includes active threat type, operating mode, frequency, assets available, assets used, receiver tuning data, threats being engaged, and currently running automatic performance evaluations. Color coding, shape coding, and varying line structures assist in rapid data interpretation.

This approach has ramifications both for curriculum and simulator design approaches. Curricula must accommodate the changes, and simulation systems must implement them. The following discussion is confined to potential simulator system impacts. It describes methods of providing instructor control over simulator training experiences, where (1) the methods do not destroy the objectivity of the training, (2) the methods feed directly into formally stated instructional objectives, and (3) instructor-inserted changes can be automatically documented in student records.

Specific Instructor Control Features. First is the capability to limit the tactical "competence" of modeled threats, both before and during the exercise. Limits can take many forms without overtly obstructing the realism or training value of the experience. For example, extend the minimum duration of each phase of the engagement (acquisition, tracking, shooting, etc. This provides the functional equivalent of a less proficient crew or fire control computer, weapon limitations, etc. It increases adversary response time to the student's error or failure to recognize and analyze a situation as quickly as necessary in the real world.

Second is allowing the instructor to increase threat susceptibility to applied countermeasures. This allows the student to demonstrate he has seen the threat and moved to counter it without his having to devote all of his attention to refining his applications early in his training.

Third is the provision of control of unusual engagement sequences. Disable short cut or "snapshooting" engagements (such as launching missiles out of what the student believes is purely an acquisition mode) until the student is prepared to recognize and address them.

Fourth is allowing the instructor to disable the endgame, inhibiting the adversary from launching at or destroying the student platform. This is particularly important if there is no endgame counter to teach the student.

These methods can be implemented by providing proficiency/lethality coefficients under instructor, lesson script, or adaptive control; by allowing instructor input of track inhibits and shoot inhibits for individual threats or threat groups; or by allowing instructors to set delays in data transfer times among members of a threat network. This has the effect of retarding network responses to student intrusion, inactivity, or ineptitude.

Tie-ins to Student Performance Measurement and Record-keeping. If the previously discussed capabilities are implemented, student evaluation or "grading" formats would have to change. In addition to the traditional letter or numeric values in each of the performance categories, stress or difficulty coefficients would be added. Data might appear as in the following tables. The scoring and evaluation system would automatically calculate adaptive scores and make recommendations regarding whether the student should be allowed to advance.

Figure 5: Notional presentation of on-line instructor changes to a scenario

DATA COLLECTION CATEGORY	OBJECTIVE	BASE THREAT MIX	BASE LEVEL OF DIFFICULTY	INSTRUCTOR CHANGES	REVISED LEVEL OF DIFFICULTY
1.					
2.					
3.					
4.					

Figure 6: Notional presentation of student base and adaptive scores

DATA COLLECTION CATEGORY	BASE SCORE	ADAPTIVE SCORE	ADAPTIVE SCORE NEEDED TO PASS TO NEXT TRAINING LEVEL
1.			
2.			
3.			
4.			

CONCLUSIONS

It is clear that customers are demanding more and more realistic training. It is equally clear that with decreasing defense budgets, training must become not just less expensive, but more efficient and effective. The concepts presented in this paper, if implemented, will allow us to proceed toward these goals by (1) providing both the students and instructors with efficient and detailed presentations of student performance, minimizing the time taken to identify exactly where a student is deficient; (2) improving the quality of student debriefs; and (3) identifying with high precision exactly what remediation will do the student the most good. Furthermore, it affords the opportunity to train students to a criterion in a more efficient fashion, in that the use of adaptive techniques can result in a "finished product" -- a student that meets all criteria -- in the shortest possible time. Finally, all this can be accomplished not just in a classroom or computer based training lab, but in the context of a high fidelity, highly reactive, wargaming simulator. This is the potential we dare not ignore.

RESEARCH & DEVELOPMENT TECHNOLOGY APPLICATIONS SUBCOMMITTEE

Chair
Jim Cooksey, PULAU Electronics Corporation

Deputy Chair
Randy Saunders, Hughes Training, Inc.

Members
Ben Blood, Coleman Research Corporation
Bob Catron, DynaLantic Corporation
Bob Parker, General Dynamics/Land Systems
Bud Miller, AAI Corporation
Col. Jim Blake, US Army Research Laboratory
Dennis Knott, Naval Education & Training Program
Doug Smith, HQ US Army Materiel Command
Dr. Barbara Sorensen, USAF
Dr. Renata Schmidt, GP Technologies
Dr. Robert Breaux, NAWCTSD
Gary Neuman, STRICOM
Gary Teper, Information Spectrum Inc.
James Burke, Harris Corporation
Jim E. Brown, USAF
John Potts, Naval Air Systems Command
Ken Beckman, Reflectone
Larry Brumback, Lockheed - FWC
Rick Worel, NAWCTSD
Ron Muffler, Evans & Sutherland
Tom Mastaglio, Loral Federal Systems
William Holtsman, CAE-Link Corporation

Section 6
Table of Contents
Research & Development Technology Applications Papers

A PC-BASED PHOTOGRAPHIC-QUALITY IMAGE GENERATOR FOR FLIGHT SIMULATION

Izidor C. Gertner
George Wolberg
Department of Computer Science
City College of New York New York, NY 10031

George A. Geri
George R. Kelly
University of Dayton Research Institute
Higley, Arizona 85236-2020

Byron J. Pierce
Melvin Thomas
Elizabeth L. Martin
Air Force
Armstrong Laboratory
Mesa, Arizona 85206

ABSTRACT

Conventional image generation techniques rely largely on polygon rendering techniques . We describe here a system that uses off-the-shelf hardware to realize high-end image generation. We have developed a prototype image generator based on two Intel i860 processors and a host 486-PC. This hardware performs perspective transformations, clipping, and texture mapping. Parametric surfaces are generated by fitting either a bilinear or bicubic polynomial to standard Defense Mapping Agency (DMA) terrain height data. Real-time texture mapping algorithms are then used to place realistic textures, obtained from real-world photographs, onto the terrain height map. In our implementation, a multiresolution image pyramid is used to generate properly filtered images on demand at the resolution required by the viewing geometry. A wide range of terrain data approximations is used depending on altitude. Coarse (fine) approximations are implemented for high (low) altitude flight. A multiresolution terrain pyramid is used to achieve this approximation. This pyramidal approach is embedded into our real-time texture mapping system with the use of an incremental scanline algorithm. The current prototype can generate a 256 x 256 x 8-bit image at 15 frames/second using only two i860 processors, and the algorithms scale sub-linearly with the number of processors.

ABOUT THE AUTHORS

Izidor Gertner earned his Ph.D in Electrical Engineering from the Technion-Israel Institue of Technology. He has performed extensive research on fast algorithms and parallel computation. **George Wolberg** received the Ph.D degree in Computer Science from Columbia University in 1990, and was awarded an NSF Presidential Young Investigator Award in 1991. His research interests include image processing, computer graphics, and computer vision. **George A. Geri** received a Ph.D in Physics from Rensselaer Polytechnic Institute. His current research interests include human vision and the perceptual evaluation of efficient imagery. **George R. Kelly** graduated from the University of Saskatchewan with a B.S. degree in Engineering Physics. He has eight years of experience in the development of high resolution helmet mounted displays for flight simulator training devices. **Byron J. Pierce** received a Ph.D in Psychology from Arizona State University. He is currently on exchange to the Defense and Civil Institute of Environmental Medicine in North York, Ontario and is conducting behavioral research on spatial perception and 3-D graphical displays. **Melvin Thomas** received an M.S. degree in Physics from Northern Arizona University. His current interest is the development of low-cost displays for flight simulation. **Elizabeth L. Martin** received her Ph.D from University of Arizona. Her research interests include all aspects of aviation training with emphasis on visual simulation.

A PC-BASED PHOTOGRAPHIC-QUALITY IMAGE GENERATOR FOR FLIGHT SIMULATION

Izidor C. Gertner
George Wolberg
Department of Computer Science
City College of New York /
CUNY
New York, NY 10031

George A. Geri
George R. Kelly
University of Dayton
Research Institute
Higley, Arizona 85236-2020

Byron J. Pierce
Melvin Thomas
Elizabeth L. Martin
Air Force
Armstrong Laboratory
Mesa, Arizona 85206

1. INTRODUCTION

Real-time flight simulation has traditionally required expensive graphics workstations. The goal of this work is to exploit recent advances in processor and memory speeds to design a high-end PC-based multiprocessor image generator for flight simulation. By matching computer architecture with the algorithms best suited for this application, a total system can be designed at far less cost than those commercially available. Furthermore, the use of off-the-shelf components will permit the system to benefit from the economies of scale for processor and memory technologies. The principal application targeted here is low-altitude flight over a largeterrain. Several key tasks must be addressed for this application: texture mapping, multi-resolution terrain and image data, high-quality antialiasing, high-speed geometry pipeline for processing large data sets, and load balancing. This paper describes a low-cost parallel processing solution to these tasks. In particular, we present a parallel polygon rendering algorithm for general-purpose MIMD(Multiple Instruction Multiple Data) message passing architectures. The hardware configuration consists of a host 80486 processor and several slave i860 processors. The current implementation consists of only two i860 processors but the design is scalable to more processors.

Section 2 describes the rendering process and includes a discussion of Gouraud shading and its use in a fast incremental implementation of texture mapping. A description of the algorithm and hardware configuration is given in Section 3, after a brief discussion of parallel rendering. Additional details concerning clipping and filtering are given in Section 4.

2. RENDERING

The process of generating images from abstract data models is known as *rendering*. Many different rendering techniques exist for visualizing a 3D scene. They vary from polygon rendering to ray tracing and radiosity techniques. The latter two methods offer more realism at the expense of system performance. Most commercial graphics workstations support polygon rendering only when special-purpose hardware is available. In this paper, we describe rendering, on off the shelf hardware.

Rendering Pipeline

There is a well-established pipeline for rendering 3D scenes (Foley , 1990). The elements of this pipeline include: modeling transformation, trivial accept/reject classification, illumination,viewing transformation, clipping, rasterization, and display. The first stage is responsible for traversing the 3D scene database and transforming the objects from the modeling coordinate system to the world coordinate system. This assembles all objects, each possibly modeled in different local coordinate systems, into one common world coordinate system. In order to avoid needless processing later in the pipeline, polygons that fall outside the view volume are culled. In the illumination step, contributions from each light source are evaluated for each polygon and color intensities are computed at each vertex. The viewing transformation step projects the 3D object coordinates into 2D screen coordinates. Culling is an optimization step that discards back-facing polygons that face away from the viewer. Clipping discards those parts of the projected polygons that lie outside the display screen. Rasterization converts transformed polygons into pixel color values.

It consists of three steps: scan conversion, visible-surface determination, and shading. Scan conversion determines which pixels lie in the projected polygons. Visible surface determination generally makes use of a z-buffer to store the depth at each pixel. The depth value at each pixel is used to determine whether the pixel's color information should be stored in the image (frame) buffer for subsequent display. If the new point lies closer than the pixel already stored in that position, the shading calculation is performed to determine the color. Three popular shading models include: flat, Gouraud, and Phong. Flat shading uses a uniform color to fill the polygon. Gouraud shading interpolates the color values stored at the vertices. Phong shading interpolates the vertex normals and recomputes the Phong illumination model at each pixel. Gouraud shading is most popular because it offers a good balance between realism and speed.

Gouraud Shading

ouraud shading is a popular intensity interpolation algorithm used to shade polygonal surfaces in computer graphics (Gouraud, 1971). It serves to enhance realism in rendered scenes that approximate curved surfaces with planar polygons. In addition to serving as a shading algorithm, we use a variant of this approach to interpolate texture coordinates. We begin with a review of Gouraud shading in this section, followed by a description of its use in texture mapping in the next section.

Gouraud shading interpolates the intensities all along a polygon, given only the true values at the vertices. It does so while operating in scanline order. This means that the output screen is rendered in a raster fashion, (e.g., scanning the polygon from top-to-bottom, with each scan moving left-to-right). This spatial coherence lends itself to a fast incremental method for computing the interior intensity values. The basic approach is illustrated in Fig. 1.

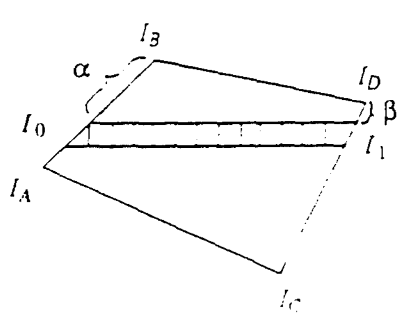


Figure 1: Incremental scanline interpolation.

For each scanline, the intensities at endpoints x_0 and and x_1 are computed. This is achieved through linear interpolation between the intensities of the appropriate polygon vertices. This yields I_0 and I_1 in Fig. 1, where

$$I_0 = \alpha I_A + (1-\alpha) I_B, \qquad 0 \leq \alpha \leq 1$$

$$I_1 = \beta I_C + (1-\beta) I_D, \qquad 0 \leq \beta \leq 1$$

Then, beginning with I_0, the intensity values along successive scanline positions are computed incrementally. In this manner, I_{x+1} can be determined directly from I_x, where the subscripts refer to positions along the scanline. We thus have

$$I_{x+1} = I_x + dI$$

where

$$dI = \frac{(I_1 - I_0)}{(x_1 - x_0)}.$$

Note that the scanline order allows us to exploit incremental computations. As a result, we are spared from having to evaluate two multiplications and two additions per pixel. Additional savings are possible by computing I_0 and I_1 incrementally as well. This requires a different set of constant increments to be added along the edges.

Incremental Texture Mapping

Although Gouraud shading has traditionally been used to interpolate intensity values, we now use it to interpolate texture coordinates. The computed *(u,v)* coordinates are used to index into the input texture. This permits us to obtain a color value that is then applied to the output pixel. The following segment of *C* code is offered as an example of how to process a single scanline.

```
dx = 1.0 / (x1 - x0);  /* normalization factor */
du = (u1 - u0) * dx;   /* constant increment for u */
dv = (v1 - v0) * dx;   /* constant increment for v */
dz = (z1 - z0) * dx;   /* constant increment for z */
```

```
for(x = x0; x < x1; x++) {
/* visit all scanline pixels */
        if(z < zbuf[x]) {
/* is new point closer? */
                zbuf[x] = z;
/* update z-buffer */
                scr[x] = tex(u,v);
/* write texture value to screen */
        }
        u += du;            /* increment u */
        v += dv;            /* increment v */
        z += dz;            /* increment z */
}
```

The procedure given above assumes that the scanline begins at *(x0,y,z0)* and ends at *(x1,y,z1)*. These two endpoints correspond to points *(u0,v0)* and *(u1,v1)*, respectively, in the input texture. For every unit step in *x*, coordinates *u* and *v* are incremented by a constant amount, e.g., *du* and *dv*, respectively. This equates to an affine mapping between a horizontal scanline in screen space and an arbitrary line in texture space with slope *dv / du* (see Fig. 2).

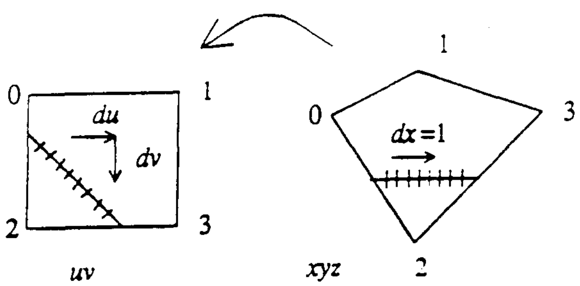


Figure 2: Incremental interpolation of texture coordinates.

Since the rendered surface may contain occluding polygons, the *z*-coordinates of visible pixels are stored in *zbuf*, the *z*-buffer for the current scanline. When a pixel is visited, its *z*-buffer entry is compared against the depth of the incoming pixel. If the incoming pixel is found to be closer, then we proceed with the computations involved in determining the output value and update the *z*-buffer with the depth of the closer point. Otherwise, the incoming point is occluded and no further action is taken on that pixel.

The function *tex(u,v)* in the above code samples the texture at point *(u,v)*. It returns an intensity value that is stored in *scr*, the screen buffer for the current scanline. For color images, RGB values would be returned by *tex* and written into three separate color channels. In the examples that follow, we let *tex* implement point sampling, e.g., no filtering. Although this introduces well-known artifacts, our goal here is to examine the geometrical properties of this simple approach. We will therefore tolerate artifacts, such as jagged edges, in the interest of simplicity. The filtering necessary for high-quality image generation is described in Section 4.

Figure 3 shows a checkerboard image mapped onto a quadrilateral using the approach described above.

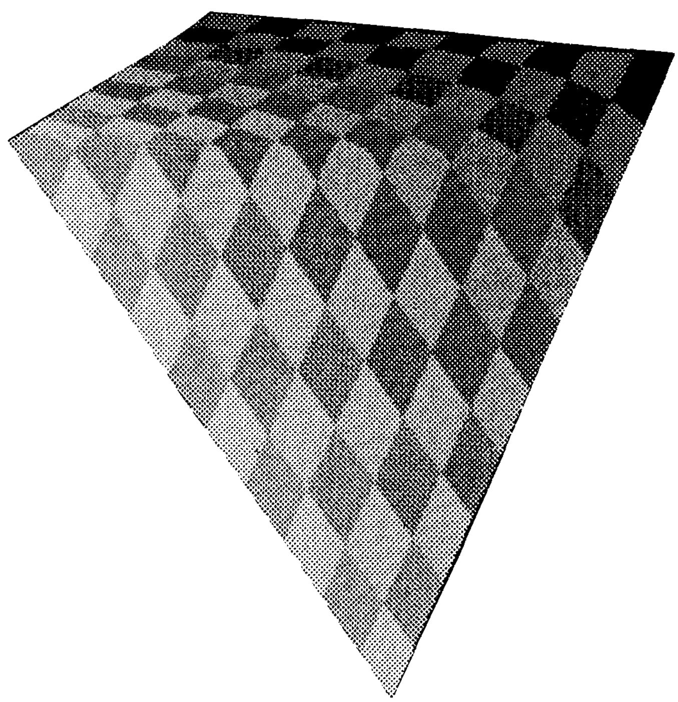


Figure 3: Naive approach applied to Checkerboard.

There are several problems that are readily noticeable. First, the textured polygon shows undesirable discontinuities along horizontal lines passing through the vertices. This is due to a sudden change in *du* and *dv* as we move past a vertex. It is an artifact of the linear interpolation of *u* and *v*. Second, the image does not exhibit the foreshortening that we would expect to see from perspective. This is due to the fact that this approach is consistent with bilinear transformation. As a result, it can be shown to be exact for affine mappings but it is inadequate to handle perspective mappings.

It is important to note that Gouraud shading has been used for years without major noticeable artifacts because shading is a slowly-varying function. However, applications such as texture mapping bring out the flaws of this approach more readily with the use of highly-varying texture patterns.

Incremental Perspective Transformations

A theoretically correct solution results by more closely examining the requirements of a perspective mapping. Since a perspective transformation is a ratio of two linear interpolants, it becomes possible to achieve theoretically correct results by introducing the divisor, i.e., homogeneous coordinate w. We thus interpolate *w* alongside *u* and *v*, and then perform two divisions per pixel. The following code contains the necessary adjustments to make the scanline approach work for perspective mappings.

```
dx = 1.0 / (x1 - x0);          /* normalization factor */
du = (u1 - u0) * dx;     /* constant increment for u */
dv = (v1 - v0) * dx;    /* constant increment for v */
dz = (z1 - z0) * dx;     /* constant increment for z */
dw = (w1 - w0) * dx;   /* constant increment for w */
for(x = x0; x < x1; x++) {  /* visit all scanline
pixels */
        if(z < zbuf[x]) {   /* is new point closer? */
                zbuf[x]  = z;  /* update z-buffer */
                scr[x] = tex(u/w,v/w);       /* write
texture value to screen */
        }
        u += du;                       /* increment u */
        v += dv;                       /* increment v */
        z += dz;                       /* increment z */
        w += dw;                       /* increment w */
}
```

Figure 4 shows the result of this method after it was applied to the Checkerboard texture. Notice the proper foreshortening and the continuity near the vertices. See (Wolberg, 1990) for a more complete discussion on the topic of real-time texture mapping and fast texture coordinate evaluation using quadratic and cubic interpolation.

3. PARALLEL RENDERING

Coupled with texture mapping, polygon rendering provides realism and visual cues for flight simulation. The chief shortcoming of polygon rendering is that it may crudely approximate the (possibly) smooth surface shape. This problem, however, can be dealt by tesselating the surface into finer (smaller) polygons. This increases the number of polygons that must be rendered per frame, thereby increasing the computational load of the image generator.

One approach to parallelizing the rendering process is to map the stages of the rendering pipeline

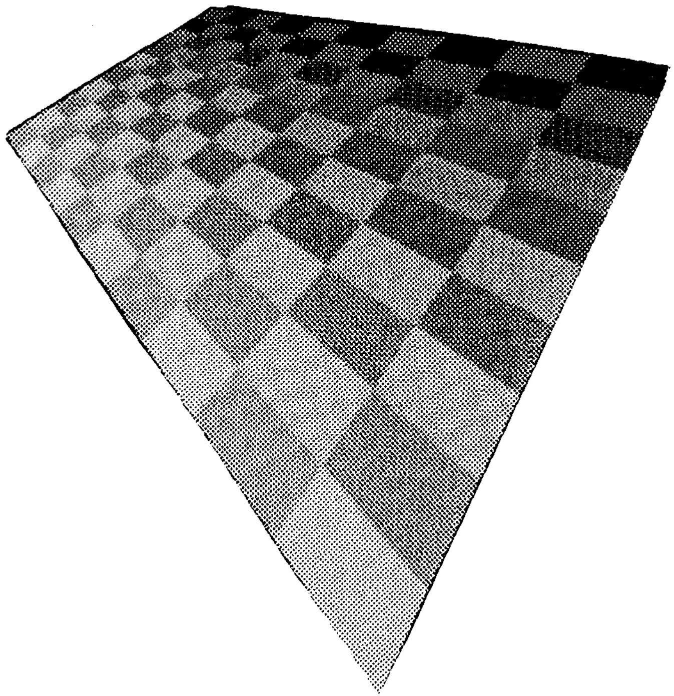


Figure 4: Perspective mapping using scanline algorihthm.

directly into hardware. In a pipeline architecture, though, the system can run only as fast as its slowest stage. Since it is generally difficult to evenly distribute the processing load over all processors, pipeline systems are not viable for rendering. Instead, we turn to a more general parallel algorithm whereby parallelism is obtained by replicating processing elements.

Close inspection of the rendering problem identifies that there are two main tasks to be parallelized: the transformation phase and the rasterization phase. Parallelism in the transformation and rasterization phases are referred to as *object parallelism* and *image parallelism* , respectively. Object parallelism is achieved by processing geometric primitives (triangles) independently of one another. Image parallelism is obtained by computing pixel values independently for individual pixels or groups of pixels.

Several groups of researchers have attacked this problem from both sides. A system with a high degree of object parallelism is described in (Torborg, 1987). The Pixel-Planes system, with a high degree of image parallelism, is described in (Fuchs, 1981). A system exploiting both object and image parallelism is described in (Molnar, 1992).

From an algorithmic standpoint, many approaches are possible. An excellent survey can be found in (Whitman, 1992). The most noteworthy systems are based on general-purpose MIMD architectures. We targeted this class of parallel architecutures because it conforms most closely with the growing trend of high-performance general-purpose processors with large instruction and data caches. In this manner, parallelism is obtained by replicating a single type of processing element. This approach is also consistent with parallelization achieved by mapping the problem onto workstations connected over a local area network.

There has been significant attention drawn to this approach in recent years. In [Barton 89], for instance, some of the issues involved in mapping the rendering pipeline onto message-passing systems are discussed. Other image and object parallelization results are presented in (Roble, 1988) and (Li, 1991). In recent work, load balancing and communication latencies in the message-passing environment are addressed in (Ellsworth, 1993). Finally, (Ortega , 1993) describes a data-parallel renderer suitable for both SIMD and MIMD architectures.

In the work described in this paper, we exploit both object and image parallelism to work on MIMD distributed-memory message-passing systems. Our rendering algorithm will run on systems containing up to p processors, where p is less than the number of scanlines in the frame buffer. Our method is unique in that it multiplexes the transformation and rasterization phases on the same processors. This has several advantages, including reduced memory utilization, overlapped computation and communication, and reduced communication contention. In addition, we ensure that all large data structures are distributed among the processors without wasteful duplication. In our case, this includes the list of polygons and the f rame buffer in which the final image is assembled. We distribute these structures evenly among the processors, allowing the algorithm to scale very complex scenes and high image resolutions by increasing the number of processors. Note that distributing the triangles corresponds to object parallelism, while distributing the image buffer corresponds to image parallelism.

Algorithm Description

The algorithm works as follows. Each of p processors is given a list of polygons to render. For optimization purposes, all polygons are restricted to be triangles. This does not pose a restriction because the input scene description of the terrain consists of elevation data on a regularly spaced mesh. Each 2×2 set of nodes on the mesh is treated as two contiguous triangles. The image buffer is divided among the p processors in equal-sized horizontal strips (see Fig. 5). Each stripe contains the same number of scanlines, which results in uniform, predictable memory requirements on each processor. The optimal size of the strips is view- and scene-dependent and remains an area of research. The following strategy is followed by each processor:

1) The lighting, transformation, and clipping steps are performed by each processor on its list of triangles. This results in 2D triangles mapped into screen coordinates on the projection plane.

2) The 2D triangles are split, if necessary, into trapezoids along the local image buffer boundaries (see Fig. 5). Each trapezoid is then sent to the processor responsible for that corresponding image buffer segment.

3) Upon receiving a trapezoid, a processor rasterizes it into its local image buffer using a standard z-buffer algorithm to eliminate hidden surfaces. The real-time texture mapping algorithm described earlier is used to shade the projected triangles.

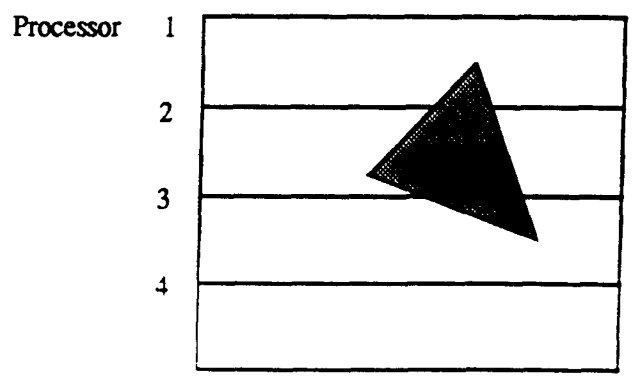


Figure 5: The image buffer is distributed across processors.

Triangles are split at boundaries. Processors start with nearly the same number of triangles, but several factors will tend to unbalance the load. This may be due to the different number of operations required to cull, clip, and subdivide the triangles. Similarly, varying the number and sizes of incoming

trapezoids will cause potentially large variations in the rasterization time.
As a result, it is best to avoid any synchronization points in the process to reduce significant amounts of idle time. Indeed, there is no synchronization point in this approach. Furthermore, to reduce communication overhead, trapezoids destined for the same processor are buffered into larger messages before sending.

Division of Labor

The host 486 processor is responsible for several tasks:

1) Monitor the joystick and update the display information on the host screen. This information includes altitude/attitude, speed, and frames per second.

2) Update the position variables for the eye in accordance with the flight dynamics.

3) Compute the M_{wv} matrix that converts all points from the world coordinate system to the viewing coordinate system. The matrix is computed asynchronously, i.e., only when the slave processors signal that they are ready to begin processing a new frame. This achieves a better sensation of real-time navigation. It is important to note that the perceived speed is equal to the product of frames/second and meters/frame. Since frames/second will likely vary, perceived speed can be made more uniform only if the matrix is not associated with equal intervals in time.

4) Distribute M_{wv} and lists of triangles to the processors.

The slave processors currently consist of two i860s. The program will be migrated to the new Analog Devices 21060 processors that are faster and support larger cache sizes, permitting larger image buffer segments to reside locally.

4. INTERPOLATION AND ANTIALIASING FILTERING

Flying over large terrain raises several interesting problems with respect to clipping and filtering. It becomes unfeasible to visit all of the triangles that comprise the terrain. Even straightforward accept/reject clipping tests becomes an overwhelming task when several million triangles must be considered at real-time rates. Instead, we propose the following solution that exploits the mesh structure of our terrain elevation data.

We project a ray from from the eye through the four corners of the view plane window. These four rays pierce the image base plane. All terrain elevation data contained in the resulting projected area are considered for viewing. This proves to be far less than the entire data set. Special care is taken when looking towards the horizon and some of the four rays may not intersect the image base plane.

Due to perspective foreshortening, we can expect many triangles to project to subpixel regions. This is particularly true for distant triangles. Such many-to-mappings give rise to undersampling, and therefore aliasing. Aliasing is a condition in which artifacts appear in the image due to undersampling the signal. A solution to aliasing is to bandlimit (blur) the image before sampling. In order to avoid having to perform blurring in real time, we preprocess the input image (texture), constructing a multi-resolution image pyramid. Those points of the texture that map into small regions are sampled from the heavily blurred pyramid level. In this manner, preprocessing the image into successively blurred levels of a pyramid permit us to avoid averaging filtering during run-time. A 17-point Hann windowed sinc function is used to build the pyramid (Wolberg, 1990). The major trade-off here is that the area of integration is approximated to be a square.

Image pyramids are of use when there is a many-to-one mapping, i.e., minification. Those points of the texture that map into large regions are magnified and must therefore be interpolated from the highest resolution pyramid level (the original). Bilinear and cubic convolution are two interpolation techniques currently supported by the system. The former (latter) method makes use of the 2×2 (4×4) set of nearest neighbors to compute the value of the pixel at a fractional position.

5. SUMMARY

We describe in this paper a PC-based photographic-quality image generator for flight simulation. In order to effectively texture map full color imagery onto high resolution terrain data at near real-time rates, a multiprocessor system has been designed to achieve both object and image parallelism. The main thrust of the algorithm is to divide the frame buffer into p horizontal strips, each associated with one of p processors. A processor rasterizes incoming triangles (or trapezoids) into its local buffer only if that triangle lies in that segment of the image. If the incoming primitive either straddles the local segment or lies entirely outside of it, the processor sends the (clipped) primitive to the appropriate processor. The original list of triangles is distributed to the p processors by a host 486 processor. Subsequently, though, the p processors pass primitives among themselves, if necessary. Each processor balances its work between clipping primitives and rasterizing them. When no more rasterization needs to be done to complete an image, the local image buffers are collected into one output frame buffer.

6. REFERENCES

Barton, E. (1989). Data Concurrency on the Meiko Computing Surface. *Parallel Processing for omputer Vision and Display.* P.M. Dew, R.A.Earnshaw and T.R. Heywood, eds., Addison-Wesley, pp. 402-407.

Ellsworth, D. (1993). A Multicomputer Polygon Rendering Algorithm for Interactive Applications. *Proceedings of the 1993 Parallel Rendering Symposium,* ACM Press, pp. 43-48.

Foley, J.D., Van Dam, A., Feiner S.K., Hughes, J.F. (1990). *Computer Graphics: Principles and Practice,* 2nd Ed., Addison-Wesley, Reading, MA.

Fuchs, H. and Poulton, J. (1981). Pixel-Planes: A VLSI-oriented design for a Raster Graphics Engine. *VLSI Design* , Vol.2, No. 3,pp. 20-28.

Gouraud, H. (1971). Continuous Shading of Curved Surfaces. *IEEE Trans. On Computers*, Vol. 20,No.6, pp., 623-628.

Li, J. and Miguet, S. (1991). Z-buffer on a Transputer-Based Machine. *Proceedings of the Sixth Distributed Memory Computing Conference.* IEEE Computer Society Press, April , pp. 315-322.

Molnar, S., Eyles, J. and Poulton, J. (1992). PixelFlow: High-Speed Rendering Using Image Composition. *Computer Graphics*, Vol. 26, No. 2, July, pp. 231-240.

Ortega, F., Hansen,C., and Ahrens, J. (1993). Fast Data Parallel Polygon Rendering. *Proceedings Supercomputing'93.* IEEE Computer Society Press, November, pp. 709-718.

Roble, D. (1988). A Load Balanced Parallel Scanline Z-Buffer Algorithm for the iPSC Hypercube. *Proceedings of Pixim'88, Paris, October* , pp. 177-192.

Torborg, J.G. (1987). A Parallel Processor Architecture for Graphics Arithmetic Operations. *Computer Graphics.* Vol. 21, No. 4, pp. 197-204.

Whitman, S. (1992). *Multiprocessor Methods for Computer Graphics Rendering.* Jones and Bartlett, Boston.

Wolberg, G. (1990). *Digital Image Warping.* IEEE Computer Society Press.

IMPLEMENTATION OF A HIGH PERFORMANCE DATABASE GENERATION SYSTEM ARCHITECTURE

Kenneth L. Merchant and Lee R. Willis
Loral Defense Systems-Akron
Akron, OH 44315-0001

ABSTRACT

As part of the Special Operations Forces Aircrew Training System, a production facility has been developed which offers a significant increase in database generation capability. This system will produce a layered, correlated database from a variety of input data sources including Defense Mapping Agency digital data, imagery, and hard copy maps. Outputs from this database are prepared for visual, infrared, or radar simulations. This system will be able to produce a 500,000 square nautical mile mission area database within 48 hours of operational tasking. This performance is made possible by a combination of state-of-the-art hardware for image and graphics processing, and specialized software tools for editing, merging, and quality control of the various data sources, and for production management.

The system is managed by a system supervisor software package which tracks available data and job resources, and allocates jobs to the workstations. An optimum schedule for processing is generated by use of a simulation model of the entire system which predicts performance based on job requirements, available resources including hardware, software, and personnel, and job execution times estimated from past performance.

A prototype workstation has been in operation at Hurlburt Field, Florida since February, 1993 with the complete system delivered in the Fall of 1994.

BIOGRAPHY

Mr. Merchant is a Section Head in the Cartographic Systems Engineering Department of Loral Defense Systems-Akron. He has over 30 years experience in the use of imagery and cartographic data for visual and sensor simulation and guidance applications. For the past four years, he has been responsible for the development of the Data Base Generation System for the Special Operations Forces Aircrew Training System (SOF ATS). Mr. Willis is also a member of the Cartographic Systems Engineering Department at Loral Defense Systems-Akron and has been responsible for the design of the internal data base and development of the system supervisor software for SOF ATS.

IMPLEMENTATION OF A HIGH PERFORMANCE DATABASE GENERATION SYSTEM ARCHITECTURE

Kenneth L. Merchant and Lee R. Willis
Loral Defense Systems - Akron
Akron, OH 44315-0001

INTRODUCTION

The Special Operations Forces Aircrew Training System (SOF ATS) being developed by Loral Defense Systems-Akron includes a highly automated production system for generation of mission rehearsal data bases. This Data Base Generation System (DBGS) offers a significant increase in data base generation capability over previous systems. It will be able to produce a 500,000 square nautical mile mission area data base within 48 hours of operational tasking from a variety of input sources. These data bases will be used by SOF aircrews to rehearse real-world missions to determine the suitability of the mission plan and prepare the aircrews for the combat environment they will be facing during the mission.

PRODUCTION CAPABILITY

Input Data Source

The SOF ATS data base production concept is based on the ability to use any of a wide variety of data types and sources in developing the data base. An interface control document (ICD) has been prepared and approved by the SOF ATS Program Office and by source data agencies which defines which sources may be made available by the government. These include 1) standard Defense Mapping Agency (DMA) digital data sets such as DFAD, DTED, DCW, ADRG, and ITD, 2) hard copy maps of all types, and 3) various types of hard copy and digital imagery, such as SPOT, Landsat, Reconnaissance cameras, and classified national sources. The DBGS has been designed to use all these sources so that a data base could be made from any combination of source data available.

Output Products

The output products from this system include on-line data bases for visual, infrared, and radar image generators which are networked to the DBGS. The internal data base contains all the attributes needed to support each of these simulations. Data base transformation software, which is part of the DBGS, selects the necessary attributes for each application, and reformats them into the output sent to the image generators. The system also can prepare an output in the Standard Simulator Data Base (SSDB) Interchange Format (SIF) for export to other data base generation systems.

The size of the data base which can be produced within 48 hours of tasking is a 500,000 square nautical mile background area with 3-5 high resolution targets and 60-80 waypoints. The exact number of targets and waypoints which can be prepared in 48 hours is a function of the actual source data supplied.

SYSTEM DESIGN

Hardware

The DBGS contains twenty separate computer systems in an integrated network for processing of data sources (see Figure 1). The system contains 10 workstations for processing input source data and producing the mission rehearsal data base. Five are image workstations which perform softcopy image processing functions. Five are graphics workstations which process digital cartographic data and map data. There are also two scanner workstations for input of hardcopy maps, and an image scanner for input of hard copy imagery. These workstations are all controlled by a supervisor station which schedules, controls, and monitors the entire DBGS.

The image workstations are manufactured by GDE Systems, San Diego, California, and together with the image data input and image data storage subsystems, are known as the Terrain Imagery Exploitation System (TIES). The TIES architecture is based on the Digital Imagery

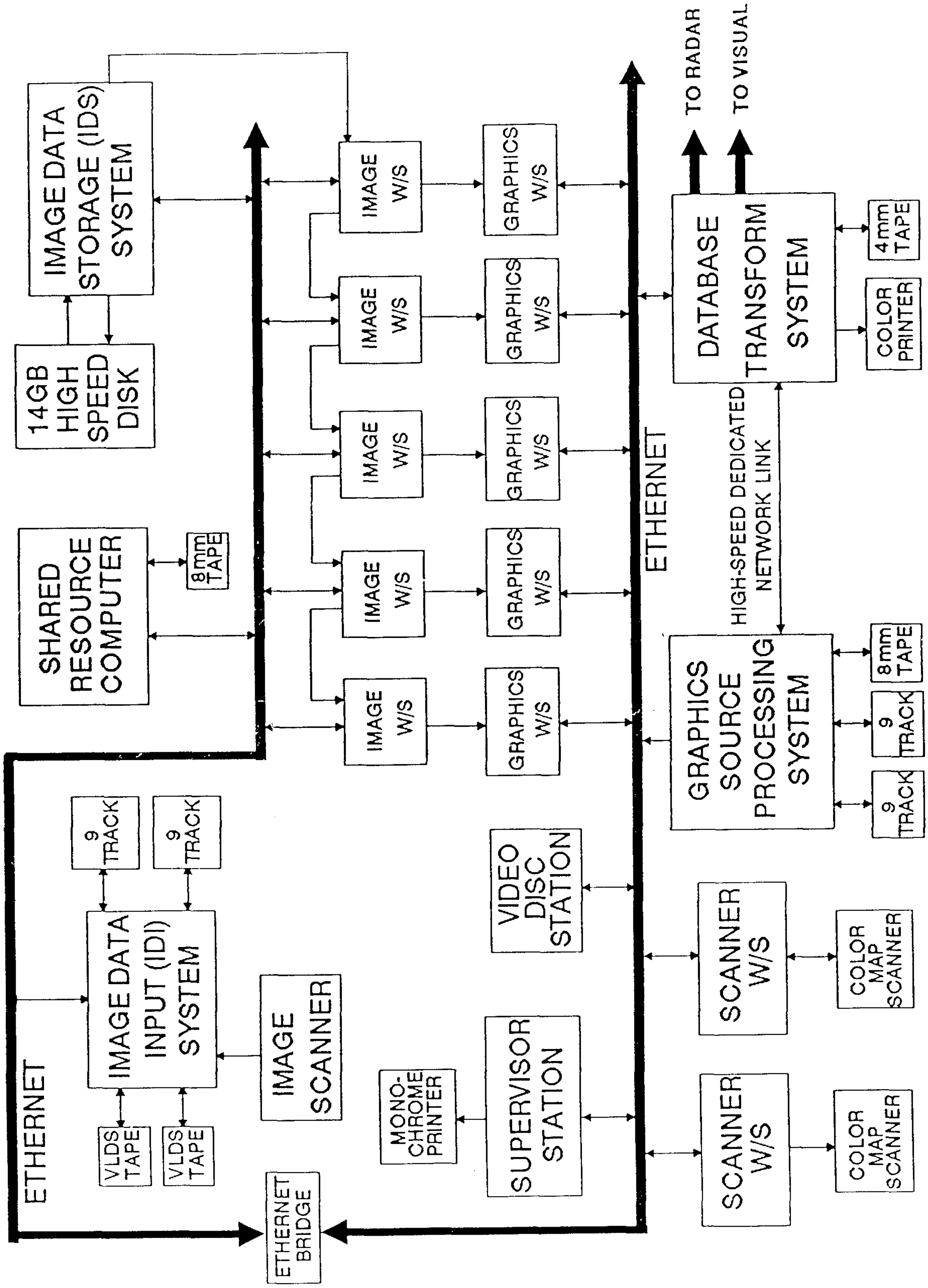


Figure 1 DBGS Configuration

Workstation Suite (DIWS) system developed for the Navy as part of the Theater Mission PlanningCenter (TMPC) upgrade. The TIES also makes use of interactive feature delineation software developed by GDE Systems, and modified for this program. The products from imagery inputs include:

a. digital terrain elevation matrices
b. 2 and 3-dimensional feature models
c. feature attributes
d. rectified image patches
e. rectified imagery mosaics

The graphics workstation consists of a Sun SPARCstation together with a digitizing table, color monitor, and stereo monitor. The function of these workstations is to assemble, edit, and perform the quality control on all components of the data base. They will have the capability to view and edit digital cartographic inputs, augment data bases from maps or imagery, produce texture and feature data from multispectral imagery, perform editing on output products from the Image Workstations, and perform a quality assessment on the final data base prior to formatting for the image generator.

The graphics source processor and data base transform processor are large Sun-compatible batch processors from Solbourne Computers which handle formatting of the input data, storage of the internal data base, and reformatting for output to the visual and sensor on-line image generators.

The Supervisor Station is also a Sun SPARCStation on which the Supervisor software for managing all the data base processing resides. This station is connected to the graphics workstations and processors by Ethernet and to the TIES Shared Resource computer by an ethernet LAN bridge. A dedicated link also connects the Graphics Source Processor and Database Transform Processor so that these batch processors can be used interchangeably.

Internal Data Base

The internal database design concept is that of a single correlated geographic data base which contains all the information necessary to support any visual or sensor simulation. The DBGS workstations process the various input sources such as maps, imagery, and digital data, and extract information required for the geographic data base. This data base includes elevation matrices, image texture data, vectors with attributes describing terrain features, and three-dimensional models of features. These components are kept in separate correlated layers within the internal data base. Editing and quality control tools have been developed for each type of data to perform such functions as feature insertion, deletion, or move, image enhancement, terrain elevation editing, and registration of the separate layers to each other. When the editing and quality control have been completed, information is extracted from the internal data base, and various transformations are applied to create specific on-line data bases for the image generators.

PRODUCTION MANAGEMENT

System Supervisor

The control of the process of building a data base and of the software used in the various processes is handled by the DBGS Supervisor software. This software includes 1) a library function for keeping track of available source data, 2) a flight profiler which sets up resolution requirements for various parts of the data base from the mission profile, 3) the Supervisor executive which controls and monitors all of the DBGS processes, 4) a performance model of all the processes in the system, which is used to allocate resources and create a job schedule for a particular data base scenario, and 5) an internal data base manager for processing data in and out of the internal data base.

Source Librarian

The Available Source Librarian maintains a compilation of all input data sources available for use during the development of the DBGS database. It is capable of updating, deleting, and listing files. The catalog data may be input by an operator or read from the input media. Attributes stored include source type, source category (feature, elevation, etc.), area of coverage, and media. This information is then supplied to the Supervisor for use in preparing a production plan.

Flight Profiler

The Flight Profiler accepts as input a mission plan with flight routes, waypoints, and targets identified by geographic coordinates. It calculates the overall extent of the data base to be built and the requirements for high detail inserts for the targets and waypoints. This information is then used by the Supervisor along with the data from the Source Librarian to create a list of jobs to build the data base. Data sources are examined in order of an established priority scheme to select the optimum sources for building the data base.

Supervisor Executive

The Supervisor Executive controls the process of building a data base and the software used in building. There are two primary functions occurring in parallel: 1) building the background or low-resolution areas, and 2) building the high resolution areas. The background is built and tiled internally on a geocell basis from DMA digital products such as DCW and/or DFAD. If a corridor exists in a geocell, the geocell is brought up to corridor resolution requirements throughout. The high resolution areas are built according to their defined boundaries regardless of geocell boundaries. Feature and elevation data for targets and waypoints will be extracted from imagery where available. In the case where insufficient images exist over a target or waypoint, the target and waypoint feature data will be created from other sources such as DMA digital products or maps. The last step in the geocell build is to merge the high resolution areas into the background.

For a typical 48 hour scenario, there may be 1000 to 2000 separate jobs, depending on the combination of source data being used, to be processed by the system. In order to process this number of jobs efficiently, a prioritized schedule is needed. The Supervisor accomplishes this by generating a graph of all production steps to be performed (see Figure 2).

This graph contains all the dependencies and prerequisites for each step. A model of the DBGS facility described in the following section is used to create an optimum schedule. The Supervisor then distributes the production jobs over the set of workstations by matching production requirements to workstation resources.

Performance Modeling

The Supervisor Performance Planner (SPP) is a simulation model of DBGS operations. The model is written using the Simulation Language for Alternative Modeling (SLAM) from the Pritsker Corp. All elements of the DBGS pertinent to generating a data base are represented in the model. A list of jobs is presented to the model along with data identifying the required information supplied by the Supervisor and simulates the allocation of each job to a suitable set of resources. Resources include all the various processors in the DBGS, memory and disk storage available on each workstation, peripherals such as tape drives and CD-ROM readers, software licenses available for commercial software (not all software resides on all workstations), and personnel available including particular job skills.

The SPP will be executed near the beginning of DBGS mission rehearsal data base production, following the definition of the data base requirements and available source materials. Within the model, an internal clock progresses so that the estimated time that each job will require its resources can be simulated; once a job is complete, the resources it was holding are freed so that another job may utilize the resources and begin execution. The model clock continues until all jobs are complete (corresponding to the completion of the DBGS data base). Through job priorities and careful selection of resources (e.g., not allocating a more powerful workstation than a job really needs while leaving another job that may require that type of workstation sitting idle), a schedule that optimizes (minimizes) the total data base generation time may be achieved. During execution of the model, the starting time and specific resources allocated to perform each job are recorded, resulting in the job schedule that is passed as output to the Supervisor Executive.

Internal Data Base Management

Data base management services for the internal data base files are provided by the Manage Internal Database (MID) software. The flow of geographic data through the DBGS involves

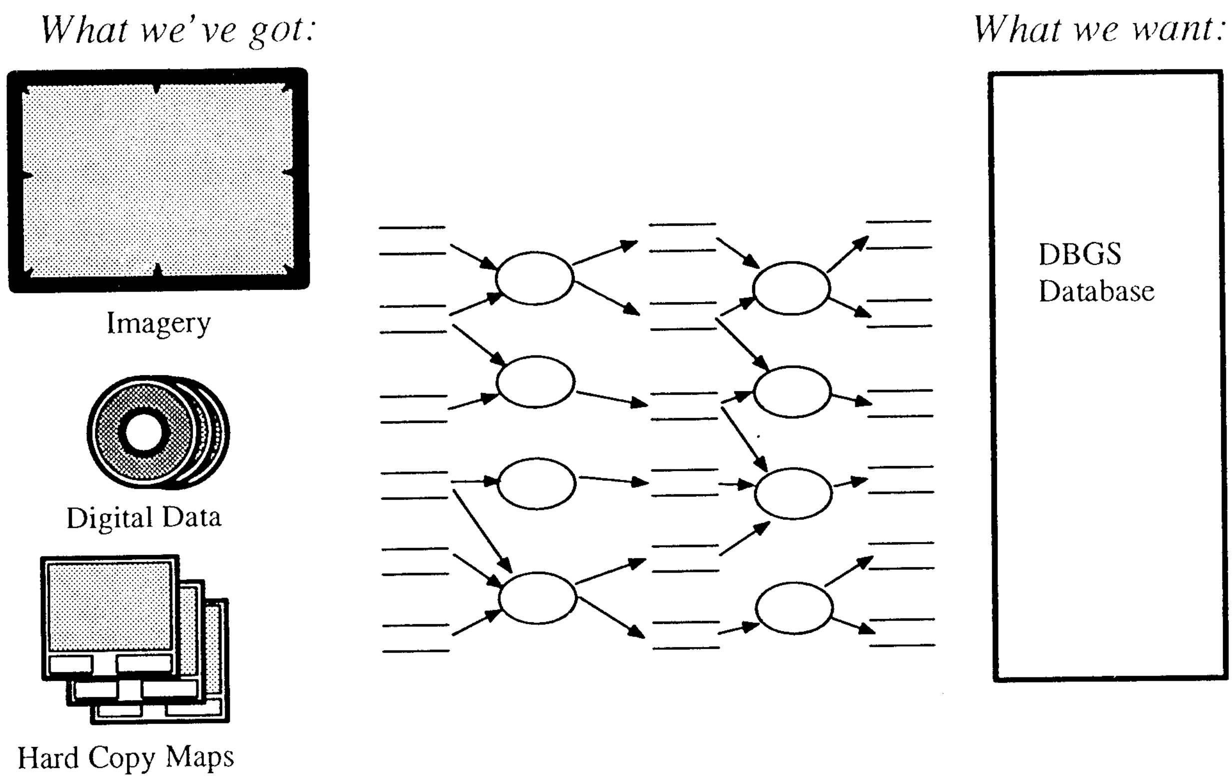


Figure 2 Supervisor Production Planning

loading, merging and editing, quality control, and output of on-line data bases. Source data are first loaded, converted into the DBGS internal format and stored in the database. From there, it is checked out for individual editing, merging, and quality control jobs at the various workstations.

MID is the only software which directly manipulates the internal data base. Other DBGS applications (e.g., Loaders, Editors) that insert or manipulate data stored in the database do so through a Supervisor controlled MID interface, and never directly access the internal data base files. Instead, copies of these data files are used. Data loading software writes its data into a temporary file which is then copied into the internal data base. Editing software receives from the MID a copy of the feature data files for the edit area. After editing, that copy is transferred back to the MID, which then copies the edited data back into the internal data base. This has the following benefits:

- This area is locked from other editing through Supervisor management, but all other areas are accessible.
- It allows the loading and editing software to perform I/O on the disk drives which are local to their process, rather than being forced to go through the network to each the disks which store the internal data base.
- It provides an error recovery path for the editing process. (e.g., if the user makes a mistake, the original data has not been lost).
- The users may edit geographic extents which do not follow the same boundaries as the source data. Internal storage is by geocell, while target areas are much smaller and may be located on the boundary between cells. Any size area and location may be checked out of the internal data base for editing.

Every time data is inserted into the internal data base where data already exists, the function is performed with an overwrite option. This means the data previously in the area will be lost. In the DBGS the Supervisor Executive takes care of the scheduling of all edit jobs and insures two users are not trying to edit the same area of the same data type at the same time.

Data Base Building

The production plan is executed under the control of the Supervisor with low-resolution background and high-resolution areas being done in parallel (see Figure 3). The Low-Res Processing steps are repeated for each geocell in the data base. The Hi-Res Processing steps are repeated for each target and waypoint. As high resolution areas are completed, they are merged into the background, thus producing a single data base which contains all the information needed for simulation. The Low-Res/Hi-Res Merge steps can occur for each geocell. These will take place when that geocell has completed its low resolution processing and all impinging high resolution areas have been completed. It is not necessary that all geocells must complete low resolution processing before any may begin merging. When a given geocell is complete, it is ready for coordinate transformation and reformatting into the output formats for either radar or visual and infrared simulation. The output data may be transmitted directly to the image generators or archived on tape for future use.

CONCLUSION

In addition to the operational advantages of providing high quality data bases for mission rehearsal, the SOF ATS DBGS offers significant productivity increases for data base production. Through the use of a system model and supervisor software for scheduling and controlling data base processes, an optimum data flow can be achieved through the system.

A prototype workstation was delivered in February, 1993 and has been in use since that time to test operational interfaces and prepare sample data bases. Development of the complete SOF ATS DBGS was achieved with the delivery of the system to Hurlburt Field, Florida in late 1994.

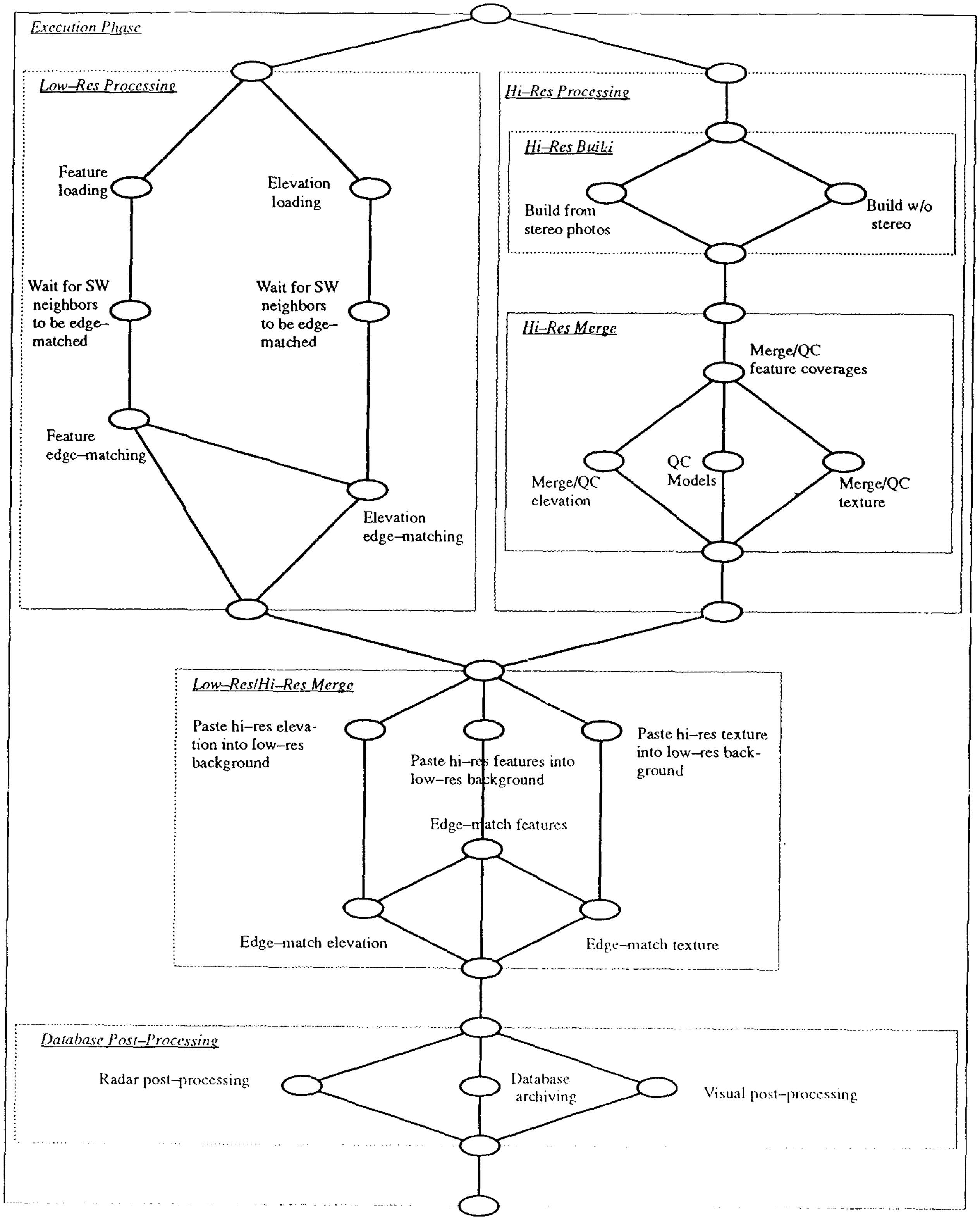


Figure 3 DBGS Processing Flow

DYNAMIC TERRAIN DATABASE DESIGN FOR REAL TIME IMAGE GENERATION

Xin Li, Dale D. Miller, Mark Illing, Mark Kenworthy and Mark Heinen
LORAL Advanced Distributed Simulation
Bellevue, Washington 98005

ABSTRACT

Substantial interest in a Dynamic Terrain (DT) database has been expressed by users and developers of real time distributed simulation and training environments. This capability allows the dynamic reconstruction of the landscape or rearrangment of the terrain surface during a simulation. One of the most challenging issues for DT in distributed simulation is the tessellation and management of the terrain database with a desired resolution meeting the real-time requirements of polygon throughput, memory allotments and interface bandwidth of the image generator.

Our research work is the first attempt of developing such capability for SIMNET image generators and databases. In this paper, the database partitioning strategies are proposed, which can be conceptually adopted by other image generators. The dynamic soil model simulating excavating activities on the terrain surface is described. The management of runtime terrain database and interface messages are presented. Implementation issues on the image generator are also discussed.

Key words: Computer image generation, real-time simulation, dynamic terrain, run-time terrain database modification, terrain relaxation, physically-based soil models.

BIOGRAPHIES

Xin Li: Dr. Xin Li is a Real-Time Software Engineer. He received his Ph. D. from the University of Central Florida and his M.S./B.S. in Computer Science from the Academic Sinica of China and the Northwest University of China. Dr. Li developed physically-based soil models while at the Institute for Simulation and Training. Since joining Loral, Dr. Li led the dynamic terrain effort described in this paper, and he is currently involved in the development of real-time rendering software based on physical-based models of clouds and smoke in a Dynamic Environment program. Dr. Xin Li can be reached at Loral ADS, 13810 SE Eastgate Way, Suite 500, Bellevue, WA 98005, (206)957-3213 (Email: xli@lads-bvu.loral.com).

Dale Miller: Dr. Dale Miller is the manager of the visual software engineering groups at Loral ADS. He received his Ph.D. in mathematics from the University of Washington in 1976. Since then he has contributed to the areas of abstract algebra, digital signal processing, applications of the residue number system for high speed digital signal processing hardware, machine vision, optical character recognition and computer graphics. Dr. Miller is the program engineer for GT200 image generator development.

Mark Illing: Mark Illing holds a B.S. in Computer Science Engineering from the University of Illinois and is currently working in the Systems Engineering Group with the real-time CIG database and development at Loral ADS. He is responsible for defining system requirements for real-time visual simulation systems, as well as designing, developing and enhancing these systems, their databases and development tools. Mr. Illing's primary focus is embedded software control of real-time hardware subsystems.

Mark Heinen: Mark Heinen received his B.S. in Computer Science from the University of Minnesota. He is currently a member of the Applications Software Group at Loral ADS. Mr. Heinen has worked on various projects including database construction tools, database compilation tools, image generation algorithms and software, CIG hardware emulation software, and CIG real-time software. He has researched and emulated image generation algorithms in software for use in the design and development of new generation CIG hardware technology.

Mark Kenworthy: Mark Kenworthy is the manager of the Systems Engineering at Loral ADS and has specialized in design and development of real time image generation systems for the last 10 years. Mr. Kenworthy holds a bachelor of Science degree in Aeronautical and Astronatical Engineering from Purdue University.

DYNAMIC TERRAIN DATABASE DESIGN FOR REAL TIME IMAGE GENERATION

Xin Li, Dale D. Miller, Mark Illing, Mark Kenworthy and Mark Heinen

1. Introduction

Previous efforts have demonstrated real-time modifications of synthetic terrain using an underlying physically-based model of the soil [Li93b]. This work has utilized a regular, fine grid for the terrain with limited extents of the virtual environment. Because of this, the total number of polygons required to represent the terrain surface remained fixed. Also, the implementation was done on a graphics workstation without textures.

The goal for the effort described in this paper was to expand upon this previous work to implement terrain modification capability on a production computer image generator (CIG) with full texturing using terrain databases of unlimited size. This required design of algorithms for real-time terrain repolygonization, texture map switching and vehicle track laydown as well as the background aggregation of polygons which preserves geometry while reducing polygon density. The repolygonization capability in turn required design of new data structures capable of representing changeable terrain. Finally, with the soil model residing on the simulation host, communication protocols between the host and the CIG were required.

This development was intended as a proof of principle, focusing on the realistic visual representation of dynamic terrain. The Loral GT100™ visual system was used for interactively bulldozing arbitrary locations on any SIMNET terrain database as shown in the image of Figure 1-1. No effort was made to attain permanence of changes or interactivity with other entities on a Distributed Interactive Simulation (DIS) network. Further work is required in networking issues and system architecture design before these new visual system capabilities can be fielded for large scale use.

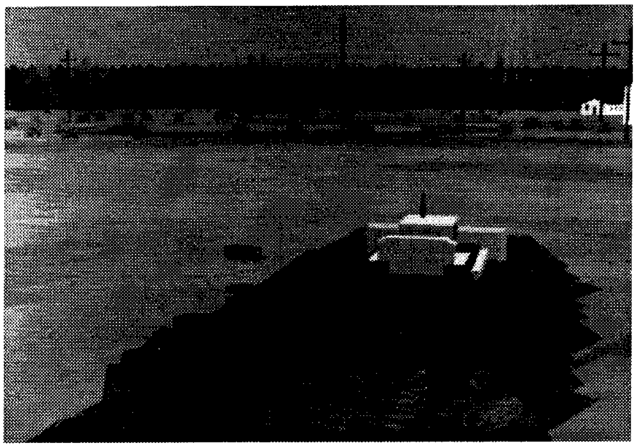


Figure 1-1 Real-time image of bulldozer modifying terrain.

2. GT100 Architecture

The GT100 is a production computer image generator (CIG) system first introduced in 1988. It is optimized for distributed (networked) interactive tactical team training in ground and near-ground vehicle applications.

The GT100 is capable of responding to the display demands of a wide variety of dynamic information that arrives in its field of view. It is designed to support the requirements of distributed simulation including very complex databases, large numbers of moving models, collision detection, correlated sensor databases, database intersection processing, and large numbers of special effects.

The GT100 system was an excellent candidate system for our first implementation of a dynamic terrain database design because the interaction of objects in the distributed simulation environment cannot be planned. The GT100 system allows a number of configurations and options to be specified by the end user. This overview of the GT100 system relates to the system used for our first dynamic terrain implementation. Complete product information for the GT100 family of image generators may be obtained through the authors.

2.1 System Overview

The major components of the GT100 visual system are shown in Figure 2.1-1. The GT100 visual system is comprised of two parts: the CIG Host Subassembly and the Graphics Processor Subassembly.

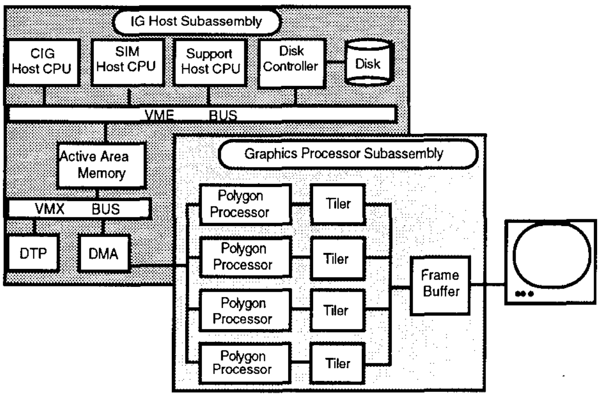


Figure 2.1-1 Dynamic Terrain Hardware Support Environment

The task of the CIG Host Subassembly is to manage efficiently the environment information so that the potentially visible three dimensional environment polygon data can be sent to the Graphics Processor Subassembly in real-time. Three processors, the CIG, SIM and Support Host CPUs, work together to place in active area memory an organized description of the simulated environment. This active area memory is simultaneously accessed by a database traversal processor (DTP) which quickly scans the environment description, determines what data needs to be sent to the Graphics Processor Subsystem, and informs the DMA controller to send that data to the Graphics Processor Subsystem.

The Graphics Processor Subassembly is a parallel pipeline graphics engine which transforms three dimensional environment polygon data into real-time video output. The data is fed to the subassembly by the CIG Host Subassembly. Each of four graphics pipelines is made up of a graphics processor and pixel tiler. Tiler output is combined and displayed by the frame buffer. (See Table 2.1-1 for GT100 system specifications.)

All of the system modifications to support the dynamic terrain demonstration were made within software modules for the CIG Host Subassembly.

2.2 Resource Utilization

The role of the three host CPUs is to manipulate the dynamic environment data in a manner which consistently provides this information to the database traversal processor (DTP/DMA). The dynamic environment manipulation task is partitioned as follows:

Table 2.1-1 GT100 System Specifications.

Image Update Rate	15 Hz
Terrain Modification Transport Delay	167 milliseconds
Textured, Anti-Aliased, Potentially Visible Polygons	90,000 polys/sec
Display Resolution	640 x 480 pixels
Pixel Fill Rate	25,000,000 pixels/sec
Occulting Levels	524, 288
Color Palette	4096
Number of Texture Maps	256
Texture Map Resolution	128 x 128 pixels

SIM Host CPU - The Simulation (SIM) Host CPU is responsible for simulating the interactions of the vehicle in the environment. In this application the vehicle is a bulldozer and it not only interacts with the terrain but also modifies the terrain. All algorithms dealing with the soil model and terrain modification execute on the SIM Host CPU.

CIG Host CPU - The CIG Host CPU is responsible for managing changes to the active area memory. As stated previously, the active area memory contains an organized description of the simulated environment which is accessed asynchronously by the database traversal processor (DTP). All requests for modifications to the terrain are managed by this processor as well as other image generator support functions.

Support Host CPU - As more and more terrain is modified by the bulldozer, a significantly large number of polygons are created that are potentially visible. The visual system has a limit to the rate at which it can process polygons. The support host is responsible for executing terrain relaxation algorithms to keep the polygon load below system thresholds.

The GT100 has a rich library of messages used to communicate between the multiple processors for simulation applications. For further detail, please refer to [CIG/SIM Comm 90].

We note here that any method for dynamic terrain on the GT100 requires additional memory than that used for a typical simulation. It is necessary for manipulating polygons and maintaining a working copy of the polygons while another copy is being displayed. Our memory utilization algorithms reuse memory when possible and we are able to run a continous exercise lasting over an hour with 1.5 MB of memory dedicated to dynamic terrain.

2.3 Terrain Format

Movement of the simulated vehicle through the environment requires paging environment data into active area memory from disk. All the data to describe a 500 meter square area is grouped together to form a load module. The active area memory has a 16 x 16 array of load modules in memory at any one time. This allows the GT100 visual system to have a viewing range of 3500 meters in any direction from any position in the database and still provide for one row or column to be in transition (paging in from disk).

Each load module contains a group of polygons that define the terrain skin. In most cases, the terrain is defined by a 4 x 4 regular tessellation with a grid spacing of 125 meters and up to 32 polygons connecting these vertices. For areas that require greater resolution than this grid supports, micro terrain provides additional terrain polygons not limited to the grid spacing.

3. Dynamic Terrain Models

This section provides a high-level description for the dynamic soil slump and manipulation models implemented for the virtual bulldozer simulation. Interested readers are referred to [Li93a] and [Li93b] for more details.

3.1 Soil Slump Model.

Given a soil configuration, e.g., a pile of soil with certain geometrical and physical properties, the soil slump model answers three questions:

1) Is the given configuration stable? (i.e., will it slump?)

2) What restoring force is required to return the soil configuration to static equilibrium if it is unstable?

3) How can mass conservation be preserved while the configuration changes state?

The stability of a given soil configuration is determined by the safety factor of a potential failure surface. According to the Mohr-coulomb theory, the safety factor is defined as a ratio of the strength force and stress force [Chowdhury 78]. The strength force provides the resistance to deformation by continuous shear displacement of soil particles along surfaces of rupture, while the stress force pushes the soil mass to move along the failure plane. If geometrical properties (area of the failure plane, volume of the soil mass) and physical properties (the cohesion, internal friction angle and unit weight of the soil) are known, both strength and stress forces can be calculated by using equations presented in [Li93b]. The configuration is statically stable if the safety factor is greater than one. Otherwise, soil sliding is inevitable.

To analyze the restoring force, the unstable soil configuration is first divided into small vertical slices with equal width as shown in Figure 3.1-1.

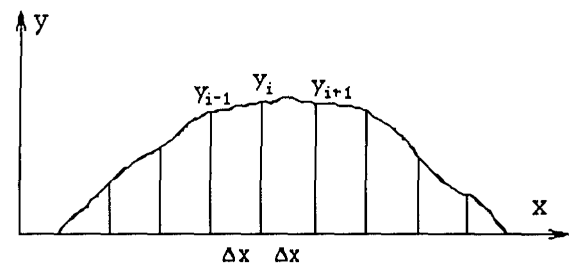


Figure 3.1-1: Dividing the given mass into small slices

The calculation of the restoring force of each slice can be done individually. Since forces exerted between each pair of soil slices are equal and in opposite directions, they can be canceled. At any particular time t, therefore, sliding can only happen in the top area of a slice. This area is further divided into slivery, v-shaped wedges and Newtonian physics is then applied to quantify net forces experienced by each wedge. The total restoring force is finally obtained by integrating small forces together [Li 93a].

Mass conservation is achieved by the following technique. Recall that a given configuration is divided into n slices. The i-th slice, $1 \le i \le n$, is now conveniently thought of as a container holding an amount of soil.

A small change of the amount of soil in each container can be viewed from two different points of view: geometrically, this change can be represented by the change of shaded area (shown in Fig 3.1-2), which is a function of the heights of soil elevation posts (e.g., y_i and y_{i+1}). On the other hand, physically, it is the amount of soil which goes out of a container, minus the amount of soil mass which goes in. This principle can be described by another function of the rate of the "flow" of soil mass, which is in terms of a function of restoring forces discussed earlier. Putting all these together, one derives n+1 ordinary differential equations with n+1 unknowns [Li 93a]. Solving these equations provides a solution for the soil slump behavior which satisfies both soil dynamics and mass conservation.

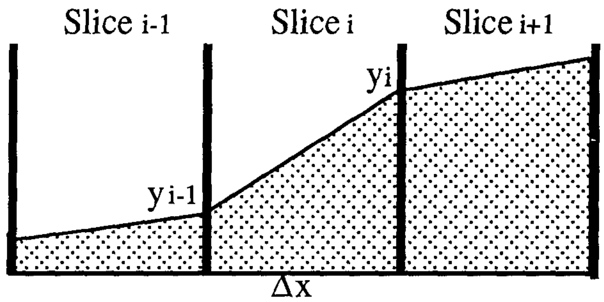


Figure 3.1-2: Considering slices as containers

3.2 Bulldozer Vehicle Dynamics Model

The bulldozer model simulates excavating activities such as digging, piling and pushing soil mass. The model is developed by first analyzing the interaction between the soil mass and a bulldozer's blade. Assuming that the shape of the blade can be approximated by an arc of a circle with radius R, we divide this arc into n segments. Furthermore, the soil mass in front of the blade is partitioned into n slices by horizontal lines at each joint point of two arc segments as shown in the following figure.

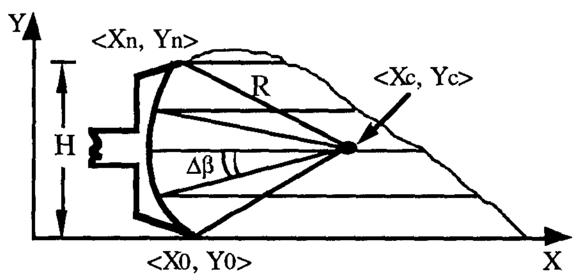


Figure 3.2-1: Dividing the blade and soil mass

If the cutting part of the bulldozer pushes the soil mass with enough force, the equilibrium will be destroyed. At this moment, the resistance experienced by each segment of the blade is determined by the soil properties (i.e., cohesion, internal and external friction angle and unit weight) and the geometry of the blade (i.e., the length and the cutting angle of the blade). Those resisting forces can be calculated for each blade segment individually by an equation presented in [Li93b]. If the force applied on a blade segment is further decomposed, we obtain two component forces: one is perpendicular to the segment, which is canceled by an opposite force provided by the blade, and another is always parallel to the surface of the blade. Integrating parallel forces of all blade segments together, we find that the total parallel force is pointing from the bottom to the top of the blade, that is, the soil mass being cut is always moving upward along the blade.

This phenomenon is also observed experimentally [Balovnev83]. The sequence of events occurring during the process of interaction between the cutting blade fixed on the advancing bulldozer and excavated soil mass before the blade can be described by three steps.

1) The soil chip being cut from the main soil mass moves upward along the blade because of resistance to the soil.

2) The soil chip is broken up into individual lumps on the upper part of the blade.

3) These lumps move downward toward the soil layers being further cut and form the soil prism which is being dragged.

Figure 3.2-2 depicts this pattern of the movement of soil mass.

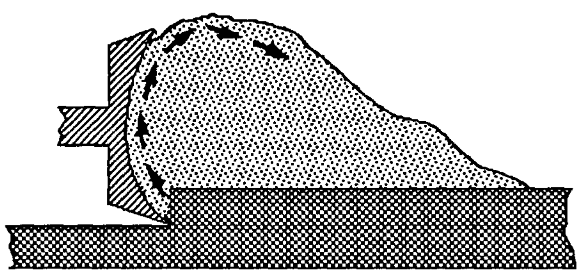


Figure 3.2-2: Pattern of soil movement ahead of the blade

This excavating action of a bulldozer is simulated by an algorithm with three stages: digging, piling and slumping. First, the model tracks the motion of the bulldozer. Along its path, wherever the altitude of soil mass is higher than the bottom of the blade, the new soil elevation is forced to have the same elevation value as the blade's bottom. This procedure will create a ditch along the path of the bulldozer on the surface of the terrain. The second stage simulates the upward movement of the soil along the blade by placing a soil chunk representing the mass cut in the first stage onto the top of the soil prism in front of the blade. Finally, in the third stage, the soil slump model introduced earlier is used to simulate the free flow motion of broken lumps of soil. Although the soil being brought to the top of the berm arrives continuously in the real world, a chunk is a reasonable approximation of the amount and location of the soil that would actually arrive during one time slice in our discrete time simulation process. The soil slump model smoothly integrates this chunk into the berm, resulting in a realistic appearance.

4. Runtime Database Modification

To manipulate terrain in real-time without visual anomalies, we developed the data services necessary to manipulate the terrain skin, implemented the prototype bulldozer simulation and reduced visual system loading with polygon reduction methods.

4.1 Terrain Manipulation Strategy

Recall that a load module is a 4x4 regular tessellation with a grid spacing of 125 meters representing the typical resolution of the terrain skin. Higher fidelity terrain can be displayed using micro terrain. Our goal was to develop a method to manipulate the terrain skin at less than 1 meter elevation post spacing for a reasonably realistic visual appearance. Replacing an entire load module with micro terrain would require over 250,000 polygons, well beyond the means of our visual system.

As a bulldozer affects only a small area around itself instantaneously, we chose to implement a hierarchical approach. Rather than replacing an entire load module with micro terrain, we progressively add detail where needed by partitioning the data into finer resolution terrain until we meet the desired resolution for manipulation. As the bulldozer moves to untouched terrain, additional partitioning occurs to allow its manipulation.

We experimented with different levels of partitioning and chose the following levels as they provide optimum data segmentation for the GT100 visual system. (See Figure 4.1-1.)

A 125 meter square of a load module is replaced with a 5x5 grid at the 1st partitioning level providing 25 meter elevation post spacing, replacing 2 polygons with 50. A square in the 1st partitioning level is replaced with a 7x7 at the 2nd level for 3.6 meter elevation post spacing, replacing 2 polygons with 98. A square in the 2nd partitioning level is replaced by a 7x7 with the 3rd and bottom partitioning level for 0.51 meter elevation post spacing replacing 2 polygons with 98 additional. Once the bulldozer is initialized and four partitioning levels are created, the majority of new changes to the partitioning merely require replacing 2 polygons from the 2nd partitioning level with 98 new polygons at the 3rd partitioning level.

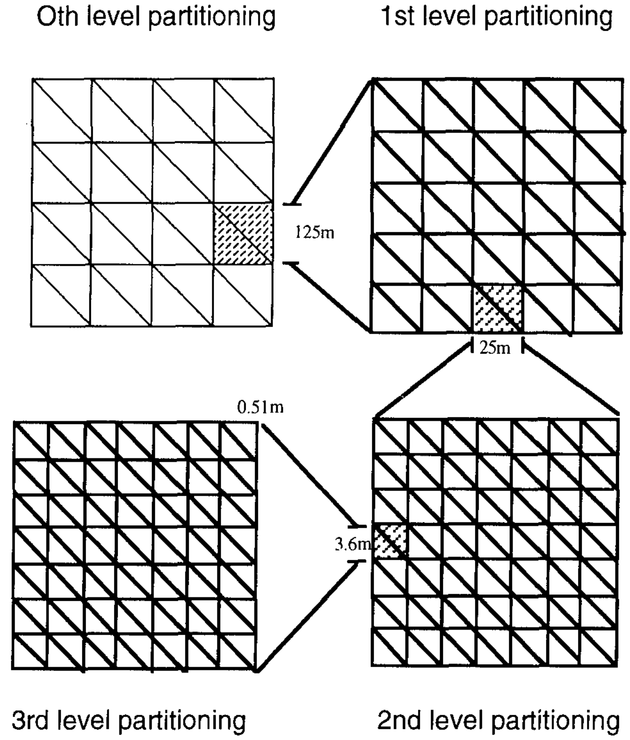


Figure 4.1-1 Terrain Partitioning

Implementation of the terrain partitioning dimensions, spacing and number of partitioning levels remains flexible, allowing tuning for a particular application or visual system. Changing the partitioning will result in coarser or finer subdivision.

In order to simplify locating and updating DT information, a data structure called a "patch" is used to represent terrain partitioning at different levels. It is an atomic unit for DT information exchange between the SIM and CIG hosts. A patch consists of three parts: geographic information, polygon graphics processor commands and database traversal processor commands which contain links to other patches at the higher, lower and same levels of terrain tessellation. Terrain patches at different partitioning levels are managed in the program by a patch forest of tree-like structures, where each root of a tree represents a load module. When a root of a patch tree is inserted into the runtime database, the geographic surface described by each patch in the tree is automatically processed and displayed by the graphics pipeline.

An example of a patch tree is demonstrated in Figure 4.1-2.

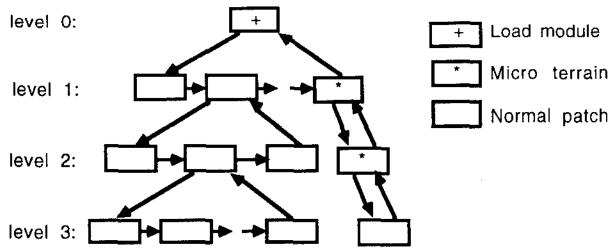


Figure 4.1-2. A Terrain Partitioning Tree

4.2 Support Services

Several new simulation support service messages were developed to provide additional functionality from the visual system host communications for the dynamic terrain implementation. These are outlined below:

MSG_DT_REQUEST is used by the simulation host to request the terrain definition for a specific location. A pointer to a bottom level partition containing the position is returned. If a partition containing the position is not found, an additional partition(s) is/are created, inserted in the processing stream and the corresponding polygons for higher level partition(s) is/are removed.

MSG_DT_PATCH is used to exchange elevation information for a modified partition between the SIM host and the CIG host. It contains the location, dimensions, spacing, and z values of elevation posts for the partition. Upon receipt of this new information, a copy of the partition is created with the new elevation values. Polygons are textured with dirt and those under the treads of the bulldozer are treated specially to display the tread pattern. It is inserted in the visual system processing stream followed by removal of the old partition. This process prevents visual anomalies of some terrain missing temporarily.

MSG_DT_RELAX is used to pass partition information between CIG host and the support host for terrain relaxation. Like MSG_DT_PATCH, it contains the position, dimensions, spacing, and z values of elevation posts for the partition. In addition, it specifies those vertices which can be referenced during the relaxation process, but are not be changed. No vertex is deleted from the list during relaxation. The relaxed patch is packed using the same message format and returned to the CIG host with a modified z value list and a new polygon list.

4.3 Excavation Activity Simulation Implementation

In this section, we define the concepts of active patch and active arena and describe the implementation of a virtual bulldozer model on the SIM host.

An active patch is a terrain patch at the bottom-level partitioning, represented by a regularly tessellated grid of equal dimension and spacing, and within a certain distance (say 5 meters) of the center of the excavating blade of a bulldozer. An active arena is an area assembled with several adjacent active patches. It is a region where elevation changes to the elevation posts of the terrain surface are likely to happen in the near future. An example of an active arena is shown in Figure 4.3-1.

6	7	8
3	4	5
0	1	2

Figure 4.3-1 An active arena of terrain excavation

During a simulation, the center of the blade is always located at the central patch (patch 4 in the figure above). When the bulldozer moves forward or backward, the blade center leaves the central patch. In order to maintain the bulldozer in the center patch, three patches are swapped out from the active arena and patch requests are issued by the SIM host. These requests are received by the CIG host and three new patches are returned to the SIM host. The active arena is then re-assembled by the SIM host. Thus, the dimensions and number of patches in the active arena remain the same.

In implementing a bulldozer model, the active arena is represented by an mxn array of elevation posts maintained by the bulldozer simulation in the SIM host. All dynamic soil computations are performed on this elevation post array. When the bulldozer moves with its blade down, the terrain surface inside the active arena is changed. Modified elevation posts are sent from the SIM host to the CIG host. The CIG host then updates the texture and vertices of polygons in the runtime terrain database in order for the changes to be viewed through the visual system. To reduce the number of messages from the SIM host to the CIG host, active patches are checked and only those with elevation post changes are sent to the CIG host.

In this approach, the data rates through the SIM host/CIG host interface per simulation frame may be very high due to new active patch data being transmitted from the CIG host to the SIM host. Recall that three patches are swapped out and three new patches are brought into the active arena when the bulldozer's blade moves across a patch boundary. During a simulation, activities of an excavating machine may coincide with a patch boundary. If the vehicle motion is oscillating across a boundary, active patches would be swapped in and out continuously, resulting a heavy network traffic.

To remedy this problem, we maintain a data structure in the SIM host to temporarily store those active patches which are just swapped out of the active arena, or likely to be used in the near future. (An algorithm was developed to predict which patches are likely to be used in the next few simulation frames.) All terrain patches brought to the SIM host are kept in a tree structure to provide not only terrain surface information required by the dynamic soil slippage and manipulation model but also geometrical data for the vehicle's terrain following. As the simulation exercise continues, the distance between some terrain patches and the center of the bulldozer's blade exceed a threshold. These patches are then discarded by the SIM host.

Maintaining some temporary storage in the SIM host increases the amount of data redundancy and causes potential data consistency problems. These drawbacks, however, are minimized by careful design and implementation. The payoff for this extra work, however, is great: the mean number of message bytes per simulation frame was reduced by two orders of magnitude.

4.4 Polygon Reduction

As discussed earlier, a load module in the run-time database is tessellated into hierarchically-structured grids at different partitioning levels when the bulldozer lays its blade down. These smaller grids create a greater polygon load for the graphics pipeline of the computer image generator. As a simulation proceeds, the number of polygons representing the fine details of the terrain surface grows. If the polygon throughput reaches a threshold, a terrain relaxation procedure is called to reduce the polygon density. In this section, we describe the terrain relaxation algorithm used for real-time relaxation of a terrain patch.

4.4.1 Relaxation Algorithm. To achieve a speed improvement in the rendering process of the image generator, the terrain relaxation algorithm is used to reduce the number of polygons required to define the terrain surface of a regularly spaced grid of elevation points. Without terrain relaxation, the surface definition consists of a list of triangles (2 triangles for each 2x2 set of elevation points). The number of triangles required to define a terrain surface for a set of nxm elevation points without terrain relaxation is: $2*(n-1)*(m-1)$.

The terrain relaxation procedure creates a list of polygons that omit those elevation points which do not add important geometrical surface information to the overall appearance of the terrain surface. An automatic polygon reduction is performed in relatively co-planar areas within a regularly spaced grid of elevation data points. A programmable tolerance value is used in the co-planarity calculation to achieve varying levels of polygon reduction, dependent upon the desired accuracy of the surface definition.

In addition, any given elevation point can be assigned to be fixed in elevation, i.e., the co-planarity calculation uses a tolerance of 0.0 to determine if that elevation point is within the plane being examined. This allows the relaxation procedure to retain some physical properties of the terrain surface, such as ridge lines or shallow ditches. Similarly, vehicle tracks or other polygons with attributes related to their appearance (color, texture or shading) are tagged to be fixed so that these features are not altered during the relaxation process.

The borders of the overall elevation grid data point set are always assumed to be fixed. This allows adjacent terrain patches at the bottom level of tessellation to be relaxed independently, but to still have an exact correspondence in their adjoining surface definition. Failure to fix the borders would create terrain separation at the patch boundaries.

4.4.2 Relaxation Strategies. There are two different strategies to determine when to relax and which terrain patches to relax: time-based strategy and distance-based strategy.

Time-based: each terrain patch at the bottom level of tessellation receives a time stamp when it is created from the terrain partitioning. It is updated with the current time whenever the patch is modified. Time stamps are routinely examined against a threshold. Those patches whose time stamp exceeds the threshold are chosen as objects for relaxation.

Distance-based: each terrain patch at the bottom level of tessellation is tagged with the distance to the center of the bulldozer's blade when it is created. This distance recalculated as the bulldozer travels. These distances are routinely compared to a threshold. Those patches whose distance exceeds the threshold are chosen as objects to relax.

In our implementation on the Loral GT100, the distance-based strategy was used. With 30 meters as the distance threshold and 0.1 meter as the relaxation tolerance, the number of polygons in the run-time database remains within a manageable range.

5. Conclusion and Future Work

A real-time interactive bulldozer simulation demonstrated dynamic terrain capability on the GT100 image generator as part of the Institute for Simulation and Training Dynamic Terrain Demonstration at I/ITSEC '93. The virtual bulldozer, driven by a Spaceball™ interface, can interactively modify any standard SIMNET terrain database at any freely-chosen location.

Fundamental problems remain before dynamic terrain capability can be fielded for large scale simulation. Even with the terrain relaxation approach used, exercises with many entities changing terrain will ultimately create so many polygons that CIGs are unable to render views and non-visual entities become computationally overburdened. New methods are needed for aggregating polygons while maintaining geometry and minimizing polygon density. Arbitration issues must also be considered for multiple entities changing a region simultaneously. Finally, the design of a system architecture which is scaleable and reliable must be addressed.

One promising method for aggregation of polygons is the use of triangulated irregular networks (TINs), which can represent the terrain relief with much lower polygonal density than can regular grids. We intend to investigate real-time methods for TINning modified terrain in future efforts.

6. Acknowledgement

The authors would like to thank the US Army Simulation, Training and Instrumentation Command (STRICOM) and the Institute for Simulation and Training (IST), who sponsored this work (contract N61339-92-K-0001).

7. References

[Balovnev83] Balovnev, V.I., *New Methods for Calculating Resistance to Cutting of Soil*, Translated from Russian, Published for the U.S. Dept. of Agriculture and the National Science Foundation, Washington, D.C., 1983.

[Chowdhury78] Chowdhury, R. N., *Slope Analysis*, Elsevier North-Holland Inc., 1978.

[CIG/SIM Comm 90] *GT100 CIG to Simulation Host Interface Manual,* BBN Systems and Technologies Corp., March 1990.

[Li93a] Li, X., *Physically-Based Modeling and Distributed Computation for Simulation of Dynamic Terrain in Virtual Environments*, Ph. D. Dissertation, University of Central Florida, March, 1993.

[Li93b] Li, X. and Moshell, J.M., "Modeling Soil: Realtime Dynamic Models for Soil slippage and Manipulation," Proceedings of SIGGRAPH '93 (Anahiem, CA, Aug. 1-6, 1993). In *ACM Computer Graphics*, vol. 27, 361-368.

INCREMENTAL REAL TIME DELAUNAY TRIANGULATION FOR TERRAIN SKIN GENERATION

Ravi Sundaram, Donald McArthur, and Venkat Devarajan
PO Box 19016, Electrical Engineering Department
University of Texas at Arlington, Arlington TX-76016

ABSTRACT

Several approaches exist in visual systems to create the terrain data bases needed to simulate flight. Terrain skin can be generated on-line by combining multiple levels of detail polygons which were originally created off-line. However, Delaunay triangulation to regenerate the terrain skin every frame time has some advantages like avoiding crack filler polygons which occur when adjacent regions are depicted in varying levels of detail. In this paper, a feasibility study is reported of the use of Dealunay triangulation in real time to regenerate the display triangles as the eye point changes. Bowyer's algorithm was used to insert new points and the Tanemura algorithm to delete points. A generic terrain model was created using fractal methods and used as input to the simulation. A time-line study using different data storage structures showed that the time taken to add a point varies $O(\sqrt{N})$ where N is the number of vertices and, the time taken to remove a point is a constant independent of the size of the current triangulation. Potential exists to reduce this to $O(N \log N)$.

ABOUT THE AUTHORS

Mr. Ravi Sundaram is currently a Development Engineer (Mesh) for Ansoft Corp., Pittsburgh, PA. After receiving a B. Tech in Aeronautical engineering from IIT Madras in 1984, he was at the Aeronautical Development Establishment, Bangalore for six years in aerodynamic design and development of unmanned aircraft. He received his Masters in Aerospace from IISc Bangalore in 1989 and joined the doctoral program of the University of Texas at Arlington in 1990. His research interests include unstructured meshes, solver development, incompressible Navier-Stokes solvers and non-Newtonian flows.

Dr. Donald McArthur is a Research Associate in the Electrical Engineering department of the University of Texas at Arlington and a senior engineering specialist at Loral Vought Systems specializing in real time simulation and image processing algorithm development. Dr. McArthur has over 16 years experience in Simulator technology. He worked at Texas Instruments in target tracking algorithms and at LTV Missiles and Electronics in real time image generation for pilot training systems and hardware-in-the-loop missile simulation. Previously, he worked in digital hardware design at Singer-Link. He received his Ph.D. at the University of Nebraska and spent nine years in teaching and nuclear physics research.

Dr. Venkat Devarajan has been an associate professor of Electrical Engineering at the University of Texas at Arlington (UTA) since March 1990 where he conducts research in visual systems technology and teaches course in Softcopy Photogrammetry and Computer Vision. Earlier, he was an Engineering Project Manager at LTV Missiles and Electronics (now Loral Vought) where he led the development of the first photo-based mission rehearsal system (TOPSCENE) and the associated data base generation system called ADAPTS for the US Navy. He has over 22 years of experience in several aspects of visual systems development. Besides performing research in this field at UTA, Dr. Devarajan has also been a visual systems technology consultant to to the US Navy, the DIA, Loral Vought Corp., McDonnell Douglas Training Systems Corp., Science Applications Int. Corp., and Hughes Training Inc.

INCREMENTAL REAL TIME DELAUNAY TRIANGULATION FOR TERRAIN SKIN GENERATION

Ravi Sundaram, Donald McArthur, and Venkat Devarajan
PO Box 19016, Electrical Engineering Department
University of Texas at Arlington, Arlington TX-76016

INTRODUCTION

Most aircraft simulators have visual systems that use computer generated imagery to display the terrain and ground features for the pilot. Proper terrain representation is usually achieved by a two fold process: a) an *off-line process* where multiple levels of detail terrain polygons are generated from terrain elevation data and stored as a part of the database and, b) a *real-time process* where the polygons of appropriate size for a given range and are chosen and projected on to a view plane. As the eye point position and orientation can change quite rapidly the image and the terrain representation must correspondingly be updated - the image at frame rates and, the polygons at some slower rate. This imposes a time limit on the number of triangles or polygons that can be processed and a polygon manager trades off between the number of polygons and the resolution of the polygons. As hardware speeds have increased more polygons per second are being processed.

The theme of this paper is that with the increased hardware speeds it is possible to avoid the expensive off line processing of polygons and have the real-time polygon processor determine the terrain representation. The decrease in the off-line data base generation process can be a major cost saver. This is especially important for mission rehearsal applications where the data base turn-around time is critical. Perhaps even more importantly, dynamic terrain representation in a distributed interactive simulation (DIS) environment will be enabled with real-time terrain representation. Thus, Incremental Delaunay Triangulation is suggested as a method of generating terrain skin in real-time. Details of the algorithm, implementation, and timing studies are presented.

THE PROBLEM DEFINITION

The terrain data is available in fine (90 m) resolution on a regular mesh. From this set of points a pruning algorithm will select a set of points, say W, which will be used for image generation. Only a subset of this W will be visible to the pilot at any instant. A set of points P_i contained in W will be designated as points on the initial triangulation. The algorithm must triangulate P_i using the Delaunay triangulation procedure. As the eye point location changes, some regions may have to be displayed with greater resolution and others with less. A few points which were invisible in the earlier view become visible and vice versa. The changes in triangle resolution are achieved by adding and deleting points for re-triangulation. (Note that the rate at which new points appear and old points disappear is not strictly tied to the frame rate of the display but is a function of the speed at which the aircraft or missile is flying, the roughness of the terrain, change in the orientation of the eyepoint etc.) So two more sets of points DP+ and DP- will be specified. The algorithm must *incrementally* add all points in DP+ and delete all points in DP- to the initial triangulation and obtain a new triangulation. This process of adding and deleting points can be indefinitely continued.

TRIANGULATION

The incremental triangulation consists of

1. an algorithm to obtain the initial triangulation for P,

2. an algorithm for adding a point (node) to the current triangulation and

3. an algorithm for deleting a point (node) from the current triangulation.

The implementation details and data structures required and timing studies for the above will be discussed.

Initial Triangulation

Consider a rectangle that would encompass all the points in W. Let a diagonal of this rectangle be joined and the resulting two triangles be the current triangulation. Any member of W would lie somewhere in the current triangulation. Since Pi is a subset of W, every point in P_i can be added to the current triangulation by calling the node insertion procedure repeatedly.

In practice, the input to this routine is the initial DTED file representing the set W. The code scans all the points in W, finds the minimum and maximum coordinates, adds four extra points called super hull points and forms the two initial triangles. Then it repeatedly calls insert_node to obtain the initial triangulation. The node insertion procedure follows.

Node Insertion Procedure

The Delaunay triangulation has many geometric properties. The most important of them is: The circumscribing circle of a triangle does not contain any other points or nodes. This is the necessary and sufficient condition for the triangulation to be Delaunay. Let T be the current triangulation and we desire to insert a point Q into the triangulation (Figure 1). The circumscribed

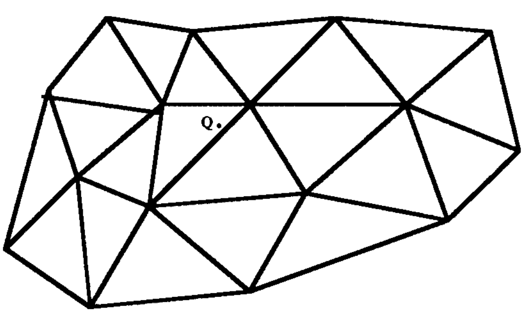


Figure 1. Initial Triangulation with the new point Q to be inserted

triangles that include the new point Q are shown in Figure 2. By the deletion of the marked triangles a cavity will be formed in

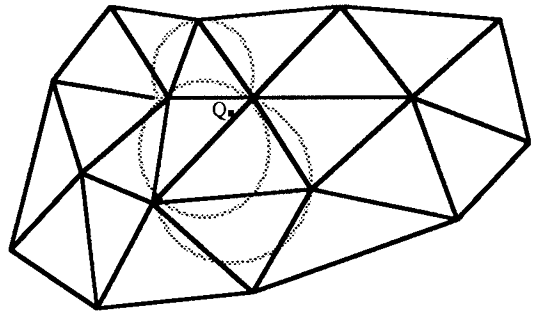


Figure 2. The circumscribed triangles that include the new point Q

the existing triangulation. The common or shared edge between a deleted triangle and its undeleted neighbor will define an edge of the cavity (Figure 3). Now the point Q could be

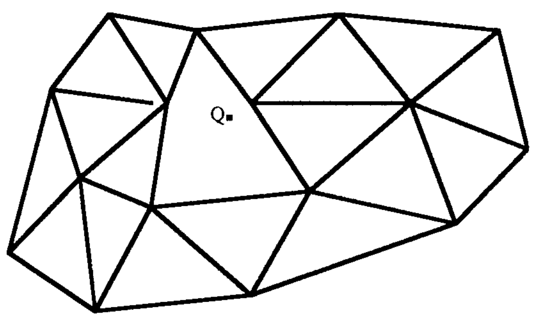


Figure 3. The cavity edges

connected to every node of the cavity edge and new triangles can be formed (Figure 4).

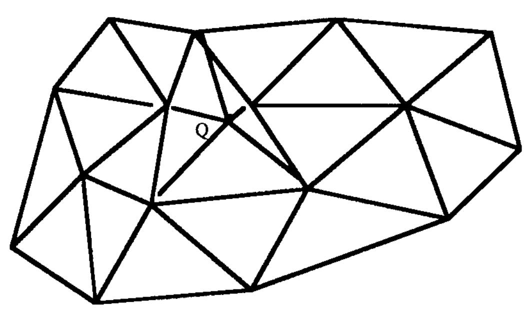


Figure 4. New Triangles formed from the cavity edges

Now the insertion is complete and the triangulation has the Delaunay property. The procedure described here is essentially due to Bowyer [2].

Node Deletion Procedure

If a node is removed then all the triangles that contain the node must be deleted. The Delaunay property [1] assures us that no other triangle has to be deleted (Figure 5). The node marked for deletion is Q.

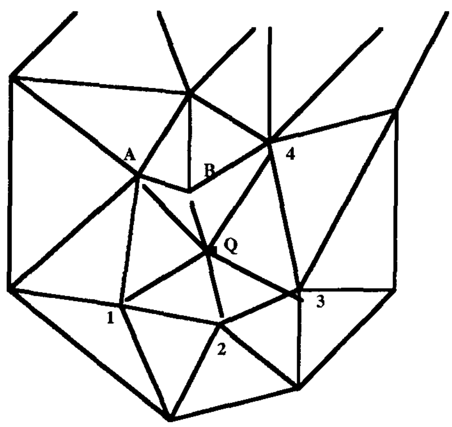


Figure 5. Triangulation with node Q to be deleted

The nodes which are connected to Q are A, B, 1, 2, 3, and 4. If node Q is deleted, the triangles ABQ, A1Q, 12Q, 23Q, 43Q and 4BQ will be deleted. Such deletion will form a cavity as shown in Figure 6. Let us call the set of nodes that form the cavity C. In this example C = 1, 2, 3, 4, A, B. Let us define an edge to be the line segment that joins any two nodes in C. Every edge connects two nodes and is shared by two triangles. If one connects a third point to the end point nodes of an edge, a triangle is formed. If such a third point lies on the non-cavity side ("wrong side") of the edge the triangle will lie, at least partially, outside the cavity. For example, connecting edge AB to node 4 will result in a triangle outside the cavity. If all the data about an edge is known, i.e., the end point nodes and the shared triangles, we can call the edge *completely known* or *complete* for short. If any piece of data is missing it will be called an *incomplete edge*. In this example, we know the end point nodes and one triangle of each edge in the cavity. The list of incomplete edges associated with the cavity is AB, B4, 43, 32, 21 and 1A. Each edge would eventually become a part of a triangle. So we take an edge, form possible triangles with each of the remaining members of C, and find the one that satisfies the Delaunay criterion.

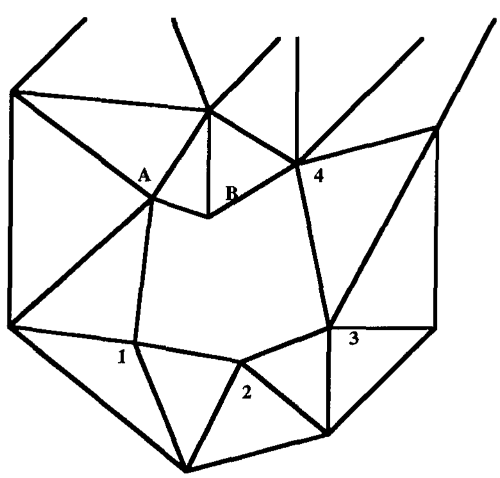


Figure 6. Cavity caused by deletion of node Q

Take for example, the edge AB. Node 4 lies on the wrong side. So three possible triangle can be formed, i.e. AB1, AB2 and AB3. If the circumcircles of these triangles are drawn it is seen that the circles through 2 and 3 include other nodes and hence violate the Delaunay criterion (Figure 7). So triangle

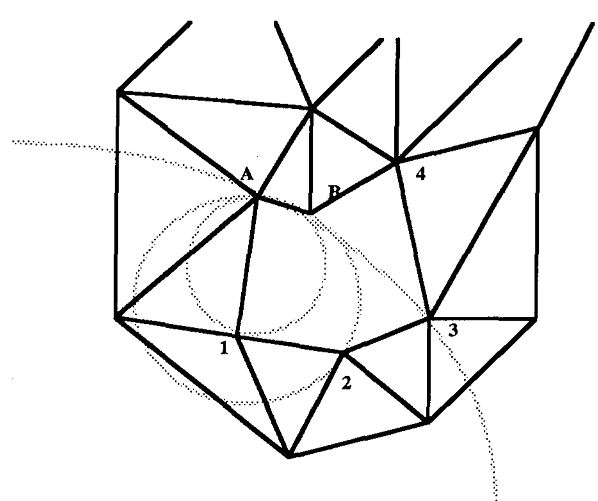


Figure 7. Set of circumcircles for edge AB and nodes 1, 2 and 3.

B1 is the correct triangle for the edge AB. Now the triangles on both sides of AB are known so this edge can be marked complete. The triangle AB1 has two more edges A1 and B1. Of the two, A1 is already present in the list of edges. So we can mark A1 complete too. The new edge B1 is added to the list of incomplete edges. This process is continued till all the edges are completely known (Figure 8).

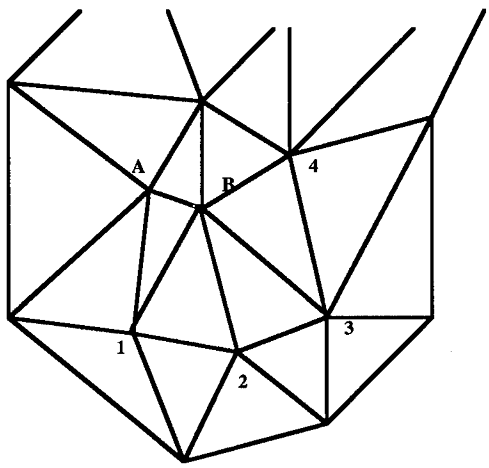


Figure 8. Result of completely filling the cavity

In practice, instead of directly checking for Delaunay Criterion an indirect testing is used based on the signed perpendicular distance of the circumcenter of the possible triangle from the edge. Exact details of the procedure and the data structures associated with them are described in the next section.

DATA STRUCTURES AND

IMLEMENTATION DETAILS

The nodes are identified by unique node numbers assigned to them. Similarly, every triangle has an unique triangle number.

The most important global data structure is the `triangle` (Table 1). The neighbors and end points are arranged in a systematic pattern. `nei[0]` would share `fp[0]` and `fp[1]` with the current triangle. Similarly, `nei[1]` would share `fp[1]` and `fp[2]` would share `fp[2]` and `fp[0]`. This makes it easy and efficient to find common edges between the neighbors.

The information stored at every node is mainly the coordinates and flags. The node information is stored in the data structure called `node_data` (Table 2).

The value of `code` would be `INSERTED` if it is part of the current triangulation It is set to `DELETED` if it is not. If this is a pseudo node inserted by the program to form the superhull this flag will have a value `SUPERHULL`. All the neighborhood and connectivity information is based on the `triangle` structure. So if we must find all the nodes connected to a particulate node we have to look up the `tri` array. It would be inefficient and prohibitively slow to search the entire array sequentially. If we know one triangle that contains the given node then we can initiate a tree search to find all triangles containing the node. For this purpose one triangle that contains the given node is stored as `handle`.

Another major structure used is `EDGE` (Table 3). This contains the data concerning edges to be used in the node deletion procedure.

The vector and the point are used to find the signed perpendicular distance of any point (x,y) from the edge. The scalar product between the vectors <nx,ny> and <(x-x1), (y-y1)> is a measure of the distance.

Other important variables are: `patch[]`, `free_list`, `ntri_valid`, `n_tri`, `n_nodes`. `patch` contains all the triangles that share a common node. As triangles are deleted their numbers are stored in `free_list`. When new triangles have to be created these numbers are popped from this list. `n_tri` is the total number of triangles used in the mesh. `ntri_valid` is the actual number of triangles in the current triangulation. This is different from `n_tri` because some of the triangles could be marked deleted.

Insert_node

The pseudo code of `insert_node` procedure is as follows:

```
insert_node(inode)
      find_one_del_tri;
      form_cavity;
      fill_cavity;
```

Table 1

```
struct triangle {
   int
         fp[3], /* Forming points. Node numbers*/
         nei[3],   /* the three neighbors.
                     Triangle numbers */
         flag;     /* if it is part of triangulation
                     this flag is CLEAR */
    double
         xcc,ycc,   /* the coordinates of circumcenter */
         cr_sqr;   /*circum_radius squared */
};
struct triangle tri[Max_TRI];
```

Table 2

```
struct node_data{
   int
       code,    /* =INSERTED or DELETED or SUPERHULL */
       handle;  /* a triangle that contains the given node */
   double
      x,x,z;  /* The coordinates */
};
struct_node_data node{MAXNODES];
```

Table 3

```
struct edge{
     int
          nf,nt,    /*Nodes from and to */
          tl,tr,     /*Triangle left and triangle right*/
          complete;   /* flag for checking completion*/
     double
          nx,ny,   /* (nx,ny) is a vector pointing to the cavity
                     side of edge  */
          x1,y1;  /* a point on edge */
};
```

When a new node has to be inserted into the triangulation all the triangles whose circumcircle contains the new node must be deleted. Searching the list of triangles sequentially is extremely inefficient. But if we know one triangle that will be deleted, then we can start a recursive search of the deleted triangle and find all of them. The first deleted triangle is found using a walk procedure. Start randomly from any node in the triangulation. Find all the triangles that surround it and check if any of them will be deleted. If not, find the node among all nodes on the patch that is nearest to the newly introduced point. Continue the walk from that node. The pseudo code for `find_one_del_tri` is shown in Table 4.

Once the triangles to be deleted are marked it is easy to form new triangles and update the data structures.

Delete_node

The pseudo code for `delete_node` is shown in Table 5.

TIMING STUDIES

The program is written in C. It is compiled using -o option in a Sparcstation. The timing details are obtained using the `gprof` program. To check the robustness of the program many input files with randomly generated points were used. The program performed flawlessly in inserting 10, 50, 100, 200, 500, 2000 and 4500 nodes.

The intent of the timing study is to find the time to insert one node into an existing triangulation (ΔT_i) and the time taken to delete one node (ΔT_d) It is found that they are typically a fraction of a millisecond. The `gprof` program has a resolution of 0.01 sec. which is inadequate for measuring fractions of milliseconds. So we inserted or deleted a large number of nodes and found the average time taken to add or delete a node. We also determined whether ΔT_i or ΔT_d depended on the size of the triangulation. In order to do this we kept the number of nodes in the current triangulation approximately the same. So a set of N points is read as the set W. N/2 points of this set were inserted and an initial triangulation obtained. Then we selected a node at random and inserted it into the triangulation if it was not present and deleted it if it was. This was repeated many times and ΔT_i and ΔT_d were determined from the cumulative times and number of calls made.

Eight input files, two with 100 nodes, two with 400, two with 900 and two with 1600 nodes were used. These nodes were generated using synthetic terrain from fractal methods. After inserting half the nodes the test procedure randomly selected nodes 20,000 times and toggled them from `INSERTED` to `DELETED` and vice versa. The input output times and other overheads were excluded and only the time taken to insert or delete nodes is reported here. The results are summarized in Figure 9 in which it is apparent that the time taken to delete a node is essentially independent of the number of nodes present in the triangulation. This is to be expected because the algorithm to delete a node only depends on the number of neighborhoods and nothing else. It is found that it takes typically 330 ± 30 microseconds to delete a node.

The time taken to insert a node depends on number of nodes. When a new node is inserted we don't know which triangle is going to be affected. So we start randomly from some node and walk towards the inserted node. The worst case is starting the walk from a diagonally opposite corner of the initial triangulation. If the nodes are assumed to be uniformly spread over the superhull we would encounter approximately $\sqrt{N}$ nodes along the way. Once a triangle to be deleted is found we follow a recursive tree search which is more efficient and is independent of the total number of nodes present in the triangulation. Figure 9 corroborates both these conclusions. The ΔT_i is seen to fall on a parabola and the time taken to find the one deleted triangle is seen to be a similar parabola shifted by a constant. The time spent on inserting a node after finding one deleted triangle is again essentially

Table 4

```
find_one_del_tri(iseed,&itiri,x,y)

      repeat{
           form_patch;      /* around iseed */
           check_patch;     /* check any one is deleted */
           if(found) return /* return if you find a deleted tri */
           find_nearest_node;  /* to node */
           iseed=nearest_node;
}forever;
```

Table 5

```
delete_node(inode)
      form_patch;  /*around inode */
                         /*form the list of edges */
      for each triangle on the patch{
            add to list of edges the side opposite to inode;
            assign the values of nf,nt,tr,nx,ny,x1,y1;
            mark this edge as incomplete;
}
form_list_of_cavity_nodes;
for each incomplete edge{
     for each node on the cavity{
         if it is on the cavity side of current edge{
               find circum center xcc,ycc;
                DIST=distance between (xcc,ycc) and edge;
                sel_node = node with lowest DIST;
              }
         }
          form a triangle using nf,nt and sel_node;
          mark this edge as complete;
          for each of other two sides of the new tri
               if already present among list of edges {
                     mark it as complete;
               }
                else{
                      add to the list of edges;
                      mark it as incomplete;
                }
          }
      }
}
```

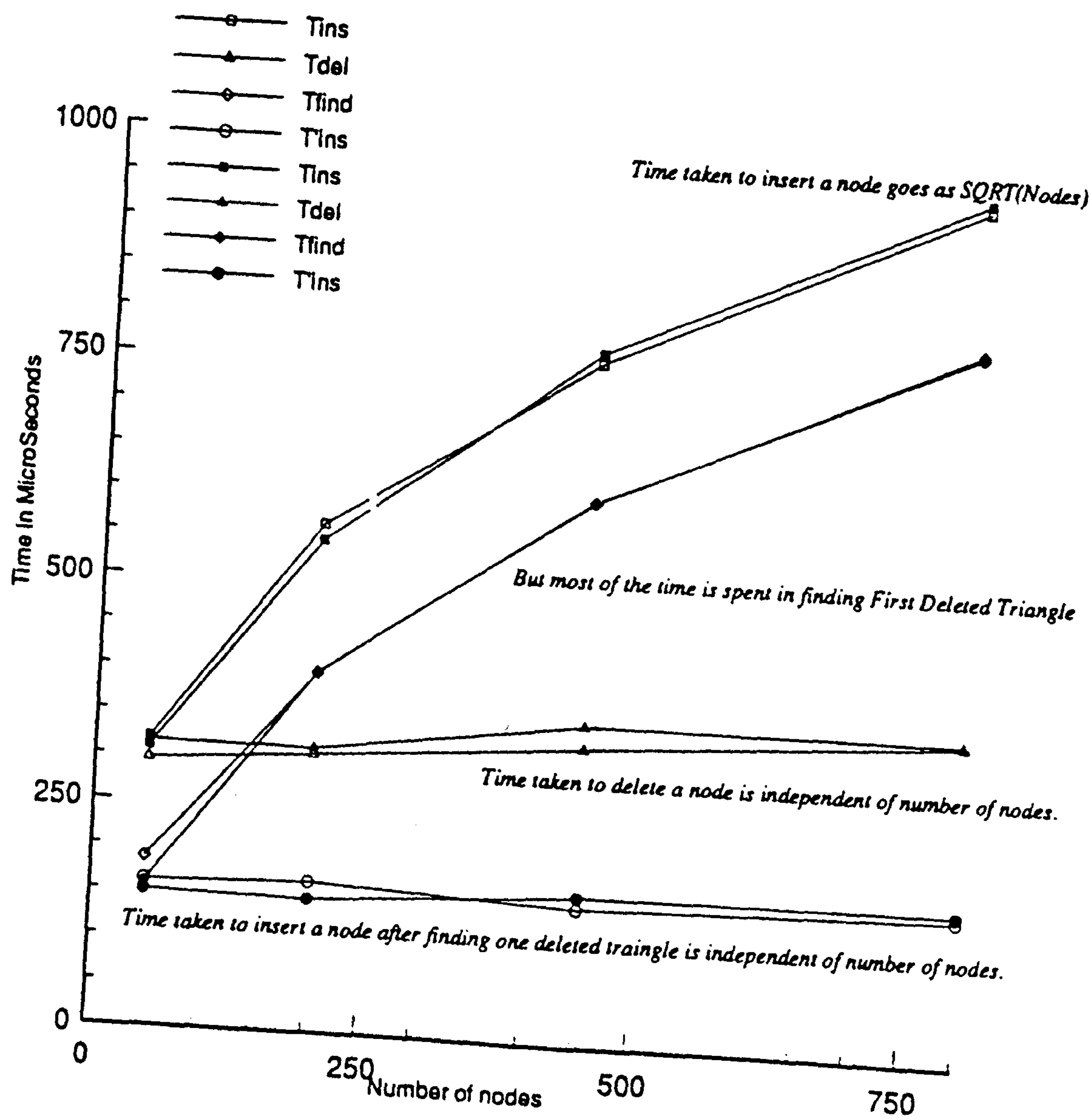


Figure 9. Timing studies for Incremental Delaunay Triangulation

independent of time. ΔTi varies between 325 µsecs in a 100 node triangulation and about 800 µsecs in an 800 node triangulation. ΔY_i generally follows the curve $\Delta T_i = 20 \sqrt{N} + 150$ µsecs. The time to insert a node after finding one deleted triangles is 155 ± 5 µsecs

In a worst case we assumed that over one second 500 nodes are deleted and are replaced with 500 new nodes. The time taken for this on a Sparcstation is seen to be 500(600+330) = 0.465 secs, leaving half a second for other functions.

CONCLUSIONS AND RECOMMENDED FUTURE WORK

Incremental Delaunay triangulation appears feasible now with the arrival of faster CPU's. The major advantages are a) Expensive off-line levels of detail generation can be eliminated and b) dynamic terrain generation is facilitated.

The next logical issue to be addressed is how to select the points to be inserted. The points to be deleted are generally the ones that get out of the instantaneous field of view of the display. Finally, the entire approach should be tested on an advanced image generator.

REFERENCES

[1] Tanemura, M., Ogawam T., and Ogita, N. .A New Algorithm for Three-Dimensional Voroni Tessellation. Journal of Computational Physics 51, pp. 191-207, (1983).

[2] Bowyer, A. Computing Dirichlet Tessellations. The Computer Journal, No. 2, Vol. 24, page 162 (1981).

NONINVASIVE MONITORING OF HELICOPTER PILOTS' INSTRUMENT SCAN PATTERNS IN A MOTION BASED SIMULATOR

P. W. Kerr, L. A. Temme, G. A. Ouellette, & D. L. Still
Naval Aerospace Medical Research Laboratory, NAS Pensacola
Pensacola, Florida, USA

The ability of a pilot to acquire and integrate information provided by the aircraft instrument console is one of the determinants of how well the aircraft is controlled. Thus, one important area of instruction of novice pilots is the selection of which instruments to attend within a given flight context and how to coordinate the information available from several instruments. However, evaluation of the effectiveness of a pilot's instrument scan is generally limited to whether he or she is able to perform a given flight maneuver successfully. Diagnosis of ineffective instrument scans and specification of remedial training would be facilitated by the knowledge of which instruments are viewed during the course of the maneuver. An eye-tracking device has been installed in a motion-based helicopter simulator at NAS Whiting Field to obtain information concerning pilot instrument scan. This device provides a noninvasive on-line video record of where a pilot is looking on the instrument panel as he or she 'flies' the simulator. In addition, flight context variables, such as instrument readings, attitude of the craft, and pilot control inputs, are time-locked to the pilot's instrument scan data and digitally recorded. A description of the noninvasiveness and accuracy of the system will be made, and pilots' and instructors' reactions to the system will be reported. A brief video tape presentation will demonstrate the information provided by the system. Plans for a connectionist modeling of the data will be described. By monitoring a pilot's eye movements in the context of the flight demands, we believe we have developed a powerful and useful research platform to study an important characteristic of pilot competence.

Paul W. Kerr is an educational and experimental psychologist who has conducted National Science Foundation and Office of Naval Research sponsored eye movement research for eight years. Dr. Kerr is a graduate of Swarthmore College where he studied cognitive psychology and linguistics. His doctoral research at the University of Illinois, Urbana-Champaign involved understanding how visual information is obtained and used during reading. Dr. Kerr is in the second year of an Office of Naval Research Postdoctoral Fellowship appointment at the Naval Aerospace Medical Research Laboratory, Pensacola, Florida.

Leonard A. Temme is a research physiologist at the Naval Aerospace Medical Research Laboratory. Dr. Temme has been on the faculty of the University of Kansas Medical School, Department of Ophthalmology, and the State University of New York at Buffalo, College of Medicine, Department of Physiology. He has held National Institutes of Health Postdoctoral Fellowships in the Department of Ophthalmology, University of Florida, Gainesville, and the Department of Psychology, Brooklyn College. He has been studying vision since he was awarded a Ph.D. in neuropsychology from the City University of New York in 1975.

Greg A. Ouellette is a student naval aviator currently assigned to VT27 in Corpus Christi, Texas. ENS Ouellette is a ROTC graduate of Purdue University from which he earned a B.S. in electrical engineering in December of 1992. He was temporarily assigned to the Naval Aerospace Medical Research Laboratory from May through October of 1993.

David L. Still is a research optometrist at the Naval Aerospace Medical Research Laboratory, and is the Navy's only O.D./Ph.D. CDR Still entered the U.S. Navy in 1975 on a health professions scholarship and began active duty as a clinical optometrist after graduating from the Illinois College of Optometry with a doctor of optometry degree in 1977. In 1984, while stationed at the U.S. Naval Hospital, Rota, Spain, he graduated from Troy State University with a masters of sciences in management and obtained a Private Pilot's license. In 1989, he successfully defended his Ph.D. dissertation in vision sciences and physiological optics, Optical Limits to Contrast Sensitivity in Human Peripheral Vision, at Indiana University. His areas of research interest include night vision goggles, unaided night vision, flight instrument displays utilizing peripheral vision, aviation vision standards, contact lenses, visual acuity, peripheral vision, night myopia, and laser eye protection.

NONINVASIVE MONITORING OF HELICOPTER PILOTS' INSTRUMENT SCAN PATTERNS IN A MOTION-BASED SIMULATOR

P. W. Kerr, L. A. Temme, G. A. Ouellette, & D. L. Still
Naval Aerospace Medical Research Laboratory, NAS Pensacola
Pensacola, Florida, USA

INTRODUCTION

Helicopter pilot instructors at the NAS Whiting Field motion-based simulator facility stated that pilot training would be facilitated if they had a means by which to view their students' instrument scan patterns as they were trained in the TH57C simulator. In response to this request, we have designed and implemented a noninvasive hardware and software system with which to view a pilot's instrument scan patterns. To minimize interference with the normal training of Navy pilots, the system is unobtrusive to the pilot, and its use and calibration have required a minimal change in training routine; yet it is sufficiently accurate, spatially and temporally, to determine which instrument the pilot is viewing at any given time.

The eye-monitoring system provides several types of video information including 1) a real-time video image of the instrument panel with a cursor overlay of the pilot's gaze position and 2) a videotape record of the instrument scan during the 'flight' which is indexed to the simulator's mission time.

In addition to the video recordings, digital records are also made of the pilot's instrument scan patterns and the simulator state information. The instrument scan information indicates the dwell durations on instruments and the order in which the instruments were viewed. The simulator state information is an exhaustive database of all aspects of the pilot's and simulator's flight behavior. Each of these data streams is time indexed to the mission time of the flight, thus allowing the investigators to obtain a comprehensive record of instrument scan in the complete context of the flight. With this comprehensive record, an analysis can be made of the relationship between the information a pilot obtains from the instrument panel and the pilot's control of the helicopter.

APPARATUS AND METHOD

Eye-Tracker Installation

Three cameras have been installed in one of instrument scan training (i.e., nonvisual) TH57C simulators at NAS Whiting Field, two of which are sensitive primarily to visible light, and the third of which is sensitive primarily to infrared light (see Figure 1). The first camera is mounted between the pilots' seats and provides a view of the instrument panel as seen by the pilot. Note that it is the pilot in the right seat of the simulator whose instrument scan is monitored. Since the instruments on the left side of the panel are not used by the pilot in the right seat, they are not imaged by camera 1.

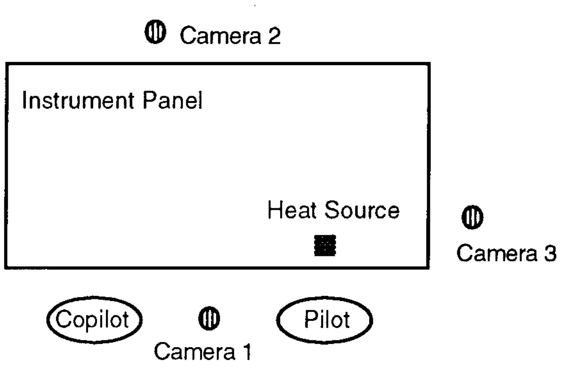


Figure 1. Sketch of the positions of the three cameras and the heat source installed in the simulator.

The second camera is mounted on the horizontal surface created by the top of the instrument panel, and faces the pilot. This camera images general activity in the cockpit, but most significantly for our purposes provides a way to see the pilot's face and to orient the third camera.

The third camera, which is sensitive to infrared (IR) light (i.e., heat, not visible to the eyes) is located right of the instrument panel (above the wet compass). It is mounted on a moveable pedestal, the orientation of which is remotely controlled. This camera images the pilot's right eye and is able to move in correspondence to the pilot's head movements to keep the right eye within the camera's view.

Background light in the cockpit is sufficient to achieve a clear picture from the first and second cameras. However, an infrared source is needed by the third camera to image the pilot's eye. A 20-Watt incandescent low-profile light bulb with an infrared pass filter has been mounted on the instrument panel . No visible light which could distract the pilot has been installed in the simulator.

It should be emphasized that the heat source and all cameras are in no way attached to the pilot (e.g., to the helmet), nor do they interfere with the pilot's or instructor's line-of-sight or operation of the helicopter simulator. With the exception of the initial moments of calibration (to be described below), we have been told frequently that the pilot and instructor quickly forget that they are being monitored soon after the training begins.

Data Collection Workstation
Output from the cameras is viewed and recorded at a computer workstation located outside the simulator. From this site the data collection operator can remotely control the eyetracking system without disturbing the simulator session or otherwise interfering with the simulator routine. The computer that is used to log the digital data is located outside the simulator as well.

Translation of Eye Reflections to Line-of-Sight Information
Two aspects of the pilot's right eye, the pupil and corneal reflection, are imaged by the IR sensitive camera and are used to compute the pilot's line-of-sight. The corneal reflection is a bright spot created by the directed heat source, identical to the reflection that can be seen from any shiny surface (e.g., a person's eyes) when it is illuminated by a visible light source. The corneal reflection serves as a fixed point of reference: the location with respect to the pilot's face at which the corneal reflection appears on the pilot's right eyeball does not move as the eye moves in its socket. However, the pupil moves as the pilot's eye rotates in direct correspondence to where the pilot is looking (see Figure 2).

These two components appear in characteristic ways as the eyes look horizontally and vertically, and are translated by a digital processing board installed in an Intel-based computer to correspond to the location on the instrument panel where the pilot is looking. However, the exact characteristics are dependent on individual differences in eye shape. Thus, for each pilot configurations of the pupil and corneal reflections at nine known locations (i.e., calibration values) must be collected as the pilot looks at specific locations within the physical region on the instrument panel he or she uses during 'flight' (see Figure 3). From these nine calibration values, all other intermediary values are computed by interpolation.

Figure 2. Representation of the relationship between the positions of the corneal reflection (CR) and pupil as the eye looks at different locations. The largest of the three circles represents the eyeball. The darkened circle is the pupil. The small circle is the CR. Note that the CR location stays the same (provided the location of the light source does not change) as the eye's gaze falls at different locations.

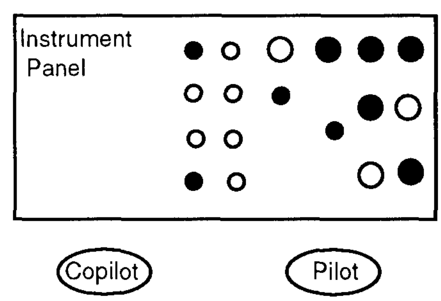


Figure 3. Location of the nine calibration points on the instrument panel. Only the right and middle sets of instruments are used by the pilot. The left set is not depicted in this figure. The circles indicate the instrument locations. The filled circles indicate the calibration points.

The IR camera, digital signal processing PC boards and translation software were manufactured by the ISCAN Corporation, Cambridge, MA, and were modified to fit the parameters of the data collection situation. According to NAMRL laboratory tests, the eye monitoring system has sufficient spatial accuracy for the task:

In an equipment configuration identical to the installation in the simulator, the eye monitoring system provided eye position data that were accurate to at least 2.4 degree of visual angle on at least 95% of the samples taken. In other words, the position of the subject's gaze as reported by the eye monitoring system falls within a circle 2.4 degrees in diameter with the subject's actual gaze position at the center of the circle. The smallest unit of interest on the instrument panel subtends approximately 4.6 degrees of visual angle when viewed from the pilot seat. The sampling rate of 60Hz is frequent enough to calculate accurate dwell durations.

Video Record

The video images from each of the three cameras are viewed in real time. The images from cameras 1 and 3 (see Figure 1) appear exactly as they are collected by the cameras. Overlaid on the image presented by the second camera, which shows the instrument panel, is a small cursor which indicates the location of the pilot's gaze at a given time.

A video tape recording is made from each video camera input. Each frame of each tape is time-stamped with the mission time for later indexing and review. All audio communication that occurs in the simulator is recorded on each tape as well.

Digital Eye Position and Simulator State Data Record

Two digital data streams are collected and integrated: the first stream is a record of where the pilot is looking on the instrument panel sampled at 60 Hz. The second stream is a record of all aspects of the simulator's state, for example the attitude and response of the craft, the readings of the instruments, the control input of the pilot (in short, any physical aspect of the pilot's flight environment). The simulator state information is a database of 479 variables sampled at 30Hz and is ported to a data logging PC via an Ethernet link. These two data streams are time-locked with reference to the mission time.

Data Collection Procedure

Data are collected on each of the 14 approximately hour-long sessions a student pilot flies in the instrument (i.e, nonvisual) trainer. As per the training schedule, a pilot flies six basic instrument (BI) flights, one emergency procedure (EP) flight, and seven radio instruments (RI) flights in the TH57C simulator with a period of training in an actual helicopter after the BI flights.

Student Pilot Selection. Pilots are selected at random to participate in the research program. Before the initial flight, a potential participant is contacted and briefed on the goals of the project and on what will be expected if he or she volunteers to take part. The pilots are informed that their participation is purely voluntary, that they can withdraw from the project at any time, that their involvement in the project will not in any way affect their career or chances of successfully completing the flight program, that the equipment cannot harm them, and that their data records will be coded to preserve anonymity.

Calibration Procedure. If a student pilot elects to participate, he or she is briefed on the following procedure prior to the first BI flight. The student sits in the pilot seat and adjusts the heat source to fully illuminate his or her face. The direction to adjust the heat source is directed by the data collection operator who, with the aid of the IR camera, can see the heat's reflection on the pilot.

The pilot is instructed to fixate in turn nine calibration locations positioned across the region of the instrument panel that the pilot views during flight. While fixating each of these locations, the data collection operator records a video sample of eye reflections that will be used as a metric to interpret all following eye position data.

To ensure that the pilot is viewing where he or she has been directed, the pilot points a low-intensity laser pencil at the specified location. The complete calibration procedure typically takes less than four minutes.

Following the initial calibration, no further requirement is made of either the student pilot or the instructor: Ater the initial calibration, the flight's procedure and training are no different in the monitored simulator from that employed for any other simulator.

Monitoring of the System. The eye-monitoring system is capable of automatically panning the eye camera to accommodate for pilot head movement and to maintain the pilot's right eye in the center of field-of-view. However, if the pilot moves abruptly or leans far from his or her original sitting position for more than approximately 2 seconds, the searching system algorithm can fail. In the event the automatic search system cannot cope with a period of pilot head movement, the data collection operator has the ability to manually reposition the eye camera.

RESULTS

Current Impact

Through the development and installation of the instrument scan monitoring device, this research project has met a need expressed by members of the Navy training community. The video portion of the system is fully operational now and is providing, or has the potential to provide, the following products to the fleet: 1) an on-line (i.e., as-it-happens) display of the pilot's scan pattern available at the instructor's workstation within the simulator, 2) an off-line playback workstation for postflight review and debrief, 3) a scan-pattern library for instruction and standardization, 4) a means by which student flight performance could be objectively evaluated, 5) a method by which changes in scan pattern because of skill acquisition and for specific maneuvers could be studied, 6) a means by which personnel selection and remediation standards could be established or improved, 7) a source of data for part-task scan trainers, 8) and, in general, a unique research facility useful to scientists interested in the study of cockpit instrument use.

We have been working closely with helicopter pilot instructors at NAS Whiting Field to maximize the usefulness of this product. The Navy flight community is very enthusiastic about the impact of this device on training, and there are plans to expand use of the device to more TH57C simulators as well as fixed-wing (T34) platforms.

Projected Impact

The next phase of the project will use the digital instrument scan data and the simulator state data for two purposes: 1) to construct a desktop part-task trainer to enable pilots to review how they scanned the instrument panel during a given motion-based simulator session and to facilitate training of more effective instrument scan, and 2) to build a computer model of instrument scan competence.

Part Task Trainer. A desktop part task trainer will be created that is driven by the two digital data streams, the simulator state information and the pilot line-of-sight information. The simulator state information will be used to replay the instrument settings during any specifiable segment of the training. The line-of-sight information will be used to position a marker that indicates where the pilot was looking at any given time during that segment.

Because both data streams are time locked and indexed to the mission time, it will be possible to instantly access and replay any portion of a given training session. In addition, a library (i.e., database) of similar maneuvers will be assembled for training purposes. For example, all instances of a pilot's instrument scans collected during a particular maneuver (e.g., all standard rate turns) could be called up for display by a pilot for independent study or by a classroom instructor.

Computer model. A computer analysis of pilots' visual and mental processing of the information provided by the helicopter instruments and how these processes change with experience has been initiated at the Naval Aerospace Medical Research Laboratory. To interpret why a pilot's eyes move from one location to another, one needs to know to what he or she is responding (i.e., what the helicopter is doing and what information is potentially available from the console at a given time). The digital information about the simulator's state will provide a context in which to interpret the scan pattern.

The resulting database of instrument scan patterns and the flight context in which they occurred is complex and difficult to understand. First, it is large and multidimensional, and therefore difficult to analyze unambiguously by traditional statistical methods. Second, subtle relationships, if present, between scan patterns and pilot competence are difficult to distinguish from the many cases that are seemingly disparate. Third, there is strong agreement that even experienced pilots are poor at introspecting from what locations and in what manner they obtain information from the instrument console. Therefore, traditional armchair analyses of video transcripts would be unreliable if accepted by themselves.

For these reasons, we prefer using a connectionist (neural network) approach. A neural net has the ability to display complicated relationships that are difficult to decipher with statistical regression and other traditional models. Moreover, the model construction process may be approached atheoretically, and thus is less apt to lead to wrong interpretations of the data: the patterns present themselves, and do not have to be programmed a priori. However, with a developed model, experimental tests and post hoc expert elaboration are still possible, and in fact necessary to tune the model.

The proposed network model will have the ability to accomplish two goals: First, for a given flight pattern, loci of instrument information will be specifiable by the model, for example, a set of instruments germane to navigation during a given maneuver. Those instruments that are viewed by expert pilots and the manner in which information from several instruments is joined (e.g. the dependence of the transitional probability from one instrument to another as a function of an instrument's reading or rate of change) will be predicted by the model. To our knowledge, no one has successfully accomplished the construction of a model of pilot competence based on real-time measures of information utilization as proposed here.

These results will be used to validate expert pilots' intuitions about their moment-by-moment processing of the instrument panel information. Predictions about the relative importance of an instrument's information during a given maneuver can be made, and the influence of other simulator state information on the use of the instrument reading can be made a priori and tested by the model.

Second, it is easy to suppose that the degree of effectiveness and efficiency with which a pilot gathers and uses the instrument information changes with experience. However, exactly how the utilization of instrument information matures is not understood. If the expert pilots' use of the instruments' information is accurately modeled, it will be possible to identify and explain the characteristics that mature with experience in the novice pilot. Thus, questions such as the following could be addressed by the model: Is it simply that each instrument is read more quickly with practice? Are interdependencies of instrument information understood more clearly with experience, and thus redundancies ignored, and only critical information viewed? Are there no substantial changes in instrument scan, but only improved visual-motor responses?

SUMMARY

In response to a request made by pilot instructors for a way to identify the locations on the instrument panel that are scanned during 'flight,' we have implemented an eye-monitoring system in a motion-based helicopter simulator at NAS Whiting Field. The eye-monitoring system meets all performance criteria, is installed within the physical constraints of the simulator, and is operational. The device provides a moment-by-moment video indication of where a pilot is looking on the simulator's instrument console. Instructors have the potential to view where the student is looking on the instrument panel during the student's simulated flight. In addition, a video record of the instrument scan is time-indexed to the mission time and videotaped for debriefing purposes.

ACKNOWLEDGEMENTS

We wish to note the significant contribution made by Mr. Bruce Bare of the Naval Air Warfare Command and Training Systems Division at NAS Whiting Field who helped to install the system described in this report.

This research was sponsored by the Naval Medical Research and Development Command and the Office of Naval Research under work unit 622233N MM33130.002-7001. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, nor the U.S. Government. Volunteer subjects were recruited, evaluated, and employed in accordance with the procedures specified in the Department of Defense Directive 3216.2 and Secretary of the Navy Instruction 3900.39 series. These instructions are based upon voluntary informed consent and meet or exceed the provisions of prevailing national and international guidelines. Trade names of materials and/or products of commercial or nongovernment organizations are cited as needed for precision. These citations do not constitute official endorsement or approval of the use of such commercial materials and/or products. Reproduction in whole or in part is permitted for any purpose of the United States Government.

THE IMPACT OF CUE FIDELITY ON PILOT BEHAVIOUR AND PERFORMANCE

Alan D White
FDS Department, Defence Research Agency
Bedford, United Kingdom

ABSTRACT

For almost as long as flight simulators have been used for pilot training, concerns have persisted that the difference in cueing environments between simulation and flight could compromise transfer of training, and therefore the training effectiveness, of synthetic devices. If these differences are intrusive then confidence in the training value of these devices will suffer and, in extreme cases, pilots may actually experience discomfort or feel sick in a way which is unrepresentative of flight. Reduced motion cues and restricted field of view are well-known differences from flight but the effects of simulator delays and harmonisation between motion and visual cues are less well understood. A knowledge of these effects is necessary if deficiencies are successfully to be countered using cue compensation techniques. Such techniques potentially offer either improved training effectiveness through better use of available cues or cheaper training devices through less-stringent cue requirements.

This paper presents the results of a study to assess the effects of inadequate and poorly-harmonised cues on pilot perception (handling qualities, workload and discomfort), pilot control behaviour and task performance. The study showed that a degraded cue environment, in the form of restricted or delayed motion and visual cues, always leads to increased workload and discomfort, modified pilot control behaviour and degraded performance. Adequate and well-harmonised cues have a major beneficial influence on pilot perception and performance, giving considerable scope for cue compensation techniques to make an impact on training effectiveness.

ABOUT THE AUTHOR

The author has worked on simulation-related activities at DRA, formerly RAE, Bedford, UK for 10 years. Work undertaken has included aircraft mathematical modelling, aircraft control law assessment, aircraft handling qualities assessment and advising the UK Ministry of Defence on training simulator issues such as operational requirements and acceptance testing. As an assignment manager within a UK MoD-funded research package he is currently engaged in identifying and refining techniques for improving the training effectiveness of piloted training simulators.

THE IMPACT OF CUE FIDELITY ON PILOT BEHAVIOUR AND PERFORMANCE

Alan D White
FDS Department, Defence Research Agency
Bedford, United Kingdom

INTRODUCTION

Background

This paper describes the results from the first in a series of trials aimed at establishing methods of making a simulator appear to pilots to fly like an aircraft in the absence of real-life motion and visual cues. These methods include cue compensation and pseudo-motion cueing. The aim of this first trial was to gain an understanding of the effects of cue disharmony in a variety of cueing environments so that future compensation schemes can be targeted where they are likely to be most effective.

Flight simulators have long been used for training pilots in aircraft handling and mission tasks. Despite this, concerns have persisted over many years about how the different sensations experienced in simulators, compared to flight, might affect pilot training[1]. These concerns arise from reports of pilots recognising discrepancies between their workload and performance of a task in flight and in simulators and, consciously or sub-consciously, modifying their control strategy to achieve more representative workload and performance in simulators. Where the training task involves an element of aircraft manoeuvring, these discrepancies have the potential to alter the workload balance in the cockpit, in some cases to the detriment of training. Where the training task involves aircraft handling as an integral part of the task, the discrepancies may result in a reduction in training value or even negative transfer of training. The psychological effect of such discrepancies on both instructors and trainees should not be under-estimated. The credibility of a training device is likely to be undermined and, in extreme cases, pilots may actually experience discomfort or feel sick in a way which is unrepresentative of flight.

Objectives

Across a wide range of training devices, from part-task trainers to high-fidelity dynamic and mission simulators, there is a need to maximise training value by optimising the effectiveness of cueing devices and by compensating for missing or false sensations. In practice, this means harmonising cueing devices so that the sensations experienced by pilots feel natural and induce aircraft-like control behaviour in response. Where insufficient cueing devices are available it may be possible to modify the simulation in a way which produces more realistic performance, control behaviour and workload balance.

Before any attempt can be made to compensate for poorly harmonised or inadequate cues, an understanding of their effects on performance, control behaviour and workload balance is required. A benchmark, against which future cue compensation or pseudo-motion techniques can be assessed, is also needed. The simulation trial described here was created to meet these requirements.

The specific objective of the trial was to quantify the effects on pilot perception, control behaviour and performance of simulator cueing deficiencies, in particular the effects of insufficient or badly-timed cues. The investigation addressed visual field of view, motion platform constraints and time delays, including the effects of unsynchronised motion and visual cues.

TRIAL PROCEDURES

Introduction

Measures of pilot perception, control behaviour and performance were used to quantify the effects of different cueing environments. Pilot perception was quantified using the Cooper-Harper handling qualities/workload rating scale[2] and a pilot discomfort scale which was created specifically for the trial. Control behaviour was quantified using stick activity measures[3] and performance was judged using objective measures of aircraft response and touchdown proficiency.

A matrix of configurations with different motion and visual delays was assessed in a variety of cueing environments. The matrix was made up of nine combinations, using three motion and three visual delays, which were presented to pilots in a random order.

The cueing environments were selected for their relevance to current and projected training requirements and were as follows :-

- fixed-base + single-window CGI visual
- fixed-base + three-window CGI visual
- conventional platform motion (synergistic platform emulation) + three-window CGI visual
- full motion cueing + three-window CGI visual (This configuration was included as a reference, as it had been validated against flight[4])

Three experienced test pilots completed an assessment of every cueing configuration in the study. Two of these were familiar with the simulator and one was not.

Simulation Environment

The trial was conducted using the Advanced Flight Simulator (AFS) facility at DRA Bedford, an important element of which is the largest motion cueing platform in Europe.

A generic fighter aircraft model was used as the baseline vehicle for the study[4]. The handling qualities of this baseline model for an approach and landing task were 'satisfactory without improvement' in the terminology of the Cooper-Harper rating scale.

The task chosen for the study was an offset approach, followed by an 'S' turn onto the runway centre-line at a height of 150 feet. This task had been used in a previous validation study[4] and was known to generate the kind of high-gain pilot behaviour which is necessary to bring out vehicle or simulator deficiencies.

Since the study aimed to provide information which could be used to assess the effectiveness of future cue compensation and pseudo-motion techniques, it was important to minimise the effects of pilot adaptation. On the other hand, the effects of adaptation were also of interest and the reliability of some measures, such as subjective ratings, was likely to improve with prolonged exposure to each configuration. By way of compromise, pilots flew two approaches for each configuration. On the first run, the reference configuration (representing flight) was flown down to the point at which the 'S' turn evaluation manoeuvre commenced, whereupon the test configuration was smoothly blended in. On the second run, the test configuration was flown for the entire approach. Pilots were encouraged to manoeuvre at the beginning of each run, either to re-acquaint themselves with the reference configuration or to practice with the test configuration.

A visual scene of the DRA Bedford airfield was generated by a Link-Miles Image 600PT CGI system with photographic texture. It was presented to the pilot on three collimated monitors with a field of view of 120° in azimuth and 30° (47° for the side monitors) in elevation. The side monitors were blanked out for single window configurations.

The Large Motion System (LMS) has three rotational and two translational degrees of freedom (heave (±5m) and sway (±4m) in this case). Ultimately, the aim of the work is to make simulators feel and perform as much like aircraft as possible. Since the AFS had already been successfully validated against flight for the approach and landing task[4], a configuration using the full capabilities of the LMS was included in the test matrix to represent the real-flight case. A second motion cueing environment was also included to emulate a conventional 6-dof synergistic platform. This involved increasing the frequencies of the motion drive 'washout' filters to constrain platform movement. Motion gains (or more accurately, attenuations) were set to be the same as the full LMS drive laws. This had the added advantage that any effects measured would be dependent on motion 'washout' frequency only. The absence of a surge degree of freedom was not considered to be significant for the approach and landing task.

GUIDE TO INTERPRETATION OF RESULTS

Measures used in the Assessment

Handling Qualities Rating: The Cooper-Harper Handling Qualities Rating[2] (HQR) scale (Figure 1) provides a subjective measure of aircraft handling qualities and piloting workload which takes into account the task performance and any pilot compensation required to achieve it. A low rating indicates that the handling qualities are satisfactory whilst higher ratings indicate degraded handling, increased workload and poorer performance.

Discomfort rating: A literature survey of reports relating to simulator-induced sickness[5-11] produced useful background material but no questionnaires or rating scales of relevance to this trial. Simulator sickness tends to occur after prolonged exposure and all the rating scales found in the literature relate to well-developed symptoms. Since pilot exposure to each configuration would be severely limited in this trial, the requirement was for a rating scale which would be sensitive to even very minor signs of discomfort. A scale, comprising the two left columns of Figure 2, was created for the trial and used with some success, though it was still not sensitive enough. If discomfort was registered by a pilot it was never greater than moderate and usually only mild with a qualifying comment. Numerical ratings, based on the scale and a review of pilot comments, were assigned later to quantify the level of pilot discomfort.

Pilot control behaviour: Pilot control activity was measured using the root mean square (rms) value and the 'pilot cutoff frequency', based on the stick force signal throughout the formal approach and landing task, ie from initiation of the 'S' manoeuvre to landing. These represent a characteristic amplitude and frequency of the command signal. The pilot cutoff frequency is a measure of pilot operating bandwidth and is defined as the frequency below which half the power in a signal is contained. The measures have been successfully validated by comparing identical vehicle configurations in the AFS and in flight[3].

Aircraft Response: The magnitude of the aircraft response was measured using the rms value of roll rate throughout the task, from 'S' turn to touchdown, to indicate how successfully pilots kept the aircraft under control.

Touchdown performance: The quality of the landing was assessed using a weighted average of several aircraft state variables at the instant of touchdown, expressed as a percentage of nominally 'ideal' touchdown values. These were landing dispersion, aircraft attitude, sink rate and airspeed. The percentages achieved are not in themselves significant: it is the variation in performance caused by changes in the cueing environment that is important. Although the measure is derived at a single point in time, it can reasonably be expected to reflect the pilots' difficulties in controlling the vehicle, provided enough measurements are taken to smooth out the inevitable variability.

Interpretation of Results

The results have been averaged for all pilots and presented as two-dimensional maps, where the horizontal axis represents added visual delay in all cases. Where a configuration includes motion cueing, the vertical axis represents added motion platform delay. Where no motion cueing is present, the vertical axis represents the (reducing) number of visual windows. The convention is that 'up' and 'right' represent a degradation in the controlled elements of the cueing environment, ie an increase in delay or a restriction in visual field of view.

As a guide to interpreting the maps, some examples are given to illustrate what the maps would look like if certain assumptions are made about the relationship between visual and motion cue delays. The maps for the fixed-base results are relatively straight-forward and need not be explained at this stage.

The simplest cases would be to assume that the variable of interest, eg pilot operating frequency or touchdown performance, is sensitive to delays in visual cueing only (Figure 3a) or in motion cueing only (Figure 3b).

If we assume that the variable is solely a function of cue disharmony, ie the difference in delay between motion and visual, then the map will look like Figure 3c. The contour lines join points where the difference in delay is the same and the value of the variable represented by each contour increases as a function of cue disharmony. In other words, the difference in the delays is more important than the absolute delays. The symmetry indicates that the function is not affected by whether motion leads visual or vice versa.

Conversely, if the variable is completely independent of cue disharmony then the contour lines will be perpendicular to the above as shown in Figure 3d. At all points on this map incremental changes to either motion or visual delay have equal effect, eg the variable may be dependent on the average of the motion and visual delays. In this case the absolute delays are more important than the differences between them.

If the variable is influenced by motion and visual delays in proportion to their absolute values then the incremental effect will be equal if the delays are equal but otherwise dominated by whichever is larger (Figure 3e). For example, changes to the motion delay will have little effect if it is small relative to the visual delay and vice versa.

DISCUSSION OF RESULTS

Introduction

Figures 4 to 6 illustrate the results for full motion cueing, conventional motion platform cueing and fixed-base cueing environments respectively. The results presented are based mainly on pilots' first runs, ie before learning could occur. The exceptions are the discomfort ratings and touchdown performance which are based on both runs. These measures required longer exposure and a greater number of samples respectively to produce sensible results. Pilot comments indicated that handling and performance was affected mainly by vehicle characteristics in the lateral axis, and this was confirmed by data analysis, so only the lateral stick activity and aircraft response are shown. Each figure contains three groups of contour plots arranged, from top to bottom, as follows:-

- o pilot perception (handling workload and discomfort)
- o pilot control activity (amplitude and frequency)
- o task performance (aircraft roll rate and touchdown proficiency).

Contour smoothing and interpolation techniques have been used to aid interpretation of the maps. General trends can be established with confidence but fine detail needs to be treated with caution because the maps have been generated from small numbers of pilots and test points.

Full Motion Cueing Environment (Figure 4)

Handling Qualities Rating: With full motion cueing the HQR map resembles Figure 3e. When closely harmonised, incremental changes in visual or motion delay have equal effect but if one delay is significantly larger than the other then incremental changes in the smaller of the two delays have little effect. An interesting difference from Figure 3e is that for small motion delays the HQRs are influenced predominantly by cue disharmony. In this area, at the bottom of the map, an increase in motion delay actually improves the HQR, ie decreases the handling workload. The data suggest that the motion platform delay is less than the visual delay, which is consistent with the simulator's known characteristics. Pilot comments confirmed that an additional motion delay improved cue harmony.

Pilot discomfort: Pilot discomfort can be seen to be almost solely a function of cue disharmony: the greater the differences in delay between visual and motion, the higher the discomfort rating (cf Figure 3c). This might be expected from current theories on simulator sickness, which link sickness to cue conflict[10]. The symmetry indicates that discomfort is independent of whether motion delay is less or more than visual delay. The bias in the vertical direction is consistent with the known difference in nominal motion and visual delays.

Pilot control behaviour: Stick magnitude increases mainly as a function of visual delay for low delays but motion delay becomes more important as the motion and visual delays increase. Motion delay does not appear to have much influence on stick frequency which decreases strongly as visual delay increases. This is surprising given the significant increase in frequency that invariably accompanies motion cues compared to visual cues alone.

Aircraft response: Aircraft roll magnitude shows much the same pattern as the pilots' stick amplitude, with motion delay becoming more influential as the visual delay increases.

Touchdown performance: The variation in touchdown performance is virtually identical to the variation in pilots' handling/workload ratings, indicating that motion and visual delays affect performance in a very similar manner to the way they affect handling qualities.

Conventional Platform Motion Cueing Environment (Figure 5)

Handling Qualities Rating: With conventional platform motion cueing the HQRs are influenced mainly by the visual delay, though large motion delays do have an effect. The small change in the slope of the contours as visual delay increases indicates an increasing tendency to be influenced by cue disharmony, an effect which is most pronounced at the lower right of the map where disharmony is greatest. The ratings are generally better than those for the full motion cueing environment, indicating that the reduced motion cues are conveying to pilots a different perception of the vehicle's handling qualities.

Discomfort rating: Like the discomfort ratings for the full motion cueing environment, those for the conventional platform motion environment are predominantly a function of disharmony. The map suggests that cue harmony would be improved by adding a considerable motion delay, much more than can be explained by the difference in nominal cue delays. The explanation probably lies in the motion drive laws, specifically the 'washout' filter which constrains the platform motions. A side effect of this filter is to distort the dynamic response of the cockpit: the higher the 'washout' frequency, the greater the phase distortion at the low frequency end of the pilots' operating spectrum. This phase lead is the opposite of the phase lag induced by time delays so, for the large-amplitude low-frequency manoeuvres likely to cause discomfort, extra time delay could compensate. The disadvantage of doing so would be poorer high frequency response, which would affect vehicle handling qualities.

Pilot control behaviour: Like the HQRs, pilot control activity is influenced predominantly by visual delay. The stick magnitudes cover a greater range than with full motion cueing and change from being dependent on absolute delays to being dependent on relative delays from top left (maximum motion delay, minimum visual delay) to bottom right (minimum motion delay, maximum visual delay). Note also that the highest control frequencies are at the top left of the map. Both these effects are consistent with the handling qualities and discomfort ratings. Pilot operating frequencies are significantly lower than with full motion cueing, probably due to the additional disharmony effects of the motion washout filtering.

Aircraft response: The aircraft roll response contours follow very similar trends to those of the stick magnitude. The roll rates generated are significantly higher than those generated with full motion cueing, even allowing for the additional stick magnitudes. This indicates that the motion cues are insufficient to induce pilots to use the same high gains that they use when full motion cueing is available.

Touchdown performance: For low visual delays, variation in touchdown performance is similar to the variation in handling quality ratings, ie incremental changes in motion delay are relatively insignificant unless the absolute motion delay is large. For large visual delays, however, motion delays have a significant effect on touchdown performance, unlike the HQRs.

Fixed-Base Cueing Environment (Figure 6)

Handling Qualities Rating: The fixed-base HQR results can most easily be compared with those for the motion cueing environments if the zero added motion delay case is used because the horizontal axis has the same meaning on all graphs. The vertical axis for the fixed-base environment represents reducing visual field of view. Along the horizontal axis the HQRs generally become poorer as we move from full motion, through conventional platform motion, to fixed-base. The three-window configuration produced a larger spread of ratings than the single-window configuration, for the same reason that motion cues normally increase the rating spread, ie the better cues show up deficiencies more clearly.

Discomfort rating: Discomfort levels for the three-window fixed-base environment are relatively low provided the delay is low but increase as the delay increases. Ratings for the single-window fixed-base environment are poor even with no added delay and extra delay makes matters even worse. These results are surprising because past evidence has suggested that wide field of view simulators are **more** likely to induce simulator sickness, not less. The answer may be that early signs of discomfort do not necessarily lead to simulator sickness or that the discomfort in this case is more psychological than physiological. A trial involving longer-duration sorties would be needed to answer this question satisfactorily.

Pilot control behaviour: Pilot stick activity shows that stick magnitude is predominantly a function of delay and that the reduction in field of view does not have a significant impact. Pilot control frequencies are reduced significantly by delay and also by restricted field of view. Compared with full motion cueing, visual delay in the fixed-base environment has a much greater influence on stick magnitude, shown by the more closely packed contours, and on stick frequencies, which are significantly lower.

Aircraft response: Field of view has only a minor effect on aircraft response magnitude compared to the effects of delay.

Touchdown performance: The restriction in field of view degrades touchdown performance. This is not reflected in the HQR results, indicating that pilots did not perceive the degradation in performance.

CONCLUSIONS

The study showed that pilots are likely to require both motion and visual cues, minimum delays and harmonised cues to achieve the same handling qualities, pilot workload, control behaviour and task performance in the simulator as in the aircraft. Inadequate or poorly-harmonised motion and visual cues degrade pilot performance.

Degraded cueing, in the form of delayed visual cues, restricted visual cues, delayed motion cues, distorted motion cues or no motion cues, causes the following:-

- increased workload as a result of degraded handling qualities
- increased pilot discomfort
- larger control movements at lower frequencies
- poorer performance

Provided the motion cues are adequate and reasonably well harmonised with visual cues, the effects of additional motion and visual delays are similar. However, if the asynchrony between visual and motion cues is large, then incremental changes in the smaller of the two delays has less effect than changes in the larger. With full motion cueing, pilot discomfort increases in proportion to motion/visual asynchrony, regardless of which cue has the bigger delay.

For conventional platform motion cueing, visual delay is the dominant influence. However, a complicating factor here is that the added motion delay counters the phase lead distortion introduced by the 'washout' filters used to constrain the platform. Discomfort ratings actually improve when motion delay is added. In general, this motion cueing environment appears to offer little advantage over the equivalent fixed-base environment and in terms of pilot operating frequency it is worse. This result **must** be treated with caution because the motion drive laws were not optimised for the vehicle.

By every measure, fixed-base cueing environments produce poorer results than a full motion cueing environment. In particular, pilot control activity is markedly greater, showing larger control amplitudes and much lower operating frequencies. Discomfort is higher than in either of the two motion cueing environments.

The study showed that an adequate, well-harmonised motion and visual cueing environment has a major beneficial influence on pilot perception, control behaviour and performance. There is therefore considerable scope for cue compensation and pseudo-motion techniques to make an impact on training effectiveness, particularly for fixed-base simulators and part-task trainers. Work is currently underway on both these techniques.

REFERENCES

1 *'Fidelity of simulation for pilot training'*, AGARD Advisory Report AGARD-AR-159, 1980.

2 Cooper G E and Harper R P, *'The use of pilot rating in the evaluation of aircraft handling qualities'*, NASA-TN-D-5153, 1969.

3 White A D, *'Measures of flight simulator fidelity based on pilot control behaviour'*, in 5th ITEC proceedings, April 1994

4 White A D, Hall J R and Tomlinson B N, *'Initial validation of an R&D simulator with large amplitude motion'*, Paper 24 in 'Piloted simulation effectiveness', AGARD-CP-513, 1991.

5 Kennedy R S, Dutton B et al, *'Simulator sickness: a survey of flight simulators for the Navy'*, SAE Paper 841597, 1985.

6 Kennedy R S, Lilienthal, M G et al, *'Simulator sickness in US Navy flight simulators'*, from Aviation, space and environmental medicine, January 1989.

7 Gower D W, Lilienthal M G et al, *'Simulator sickness in US Army and Navy fixed- and rotary-wing flight simulators'*, Paper 8 in AGARD conference on 'Motion cues in flight simulation and simulation induced sickness', AGARD-CP-433, June 1988.

8 Casali J G and Frank L H, *'Manifestation of visual/vestibular disruption in simulators: severity and empirical measurement of symptomatology'*, Paper 11 in AGARD conference on 'Motion cues in flight simulation and simulation induced sickness', AGARD-CP-433, June 1988.

9 Kennedy R S, Frank L H et al, *'Simulator sickness: reaction to a transformed perceptual world VI. Preliminary site surveys'*, Paper 34 in AGARD conference on 'Motion sickness: mechanisms, prediction, prevention and treatment', AGARD-CP-372, November 1984.

10 Griffin M J, *'Physical characteristics of stimuli provoking motion sickness'*, Paper 3 from AGARD lecture series on 'Motion sickness: significance in aerospace operations and prophylaxis', AGARD-LS-175, September 1991.

11 Money K E, *'Simulator sickness'*, Paper 6B in AGARD lecture series on 'Motion sickness: significance in aerospace operations and prophylaxis', AGARD-LS-175, September 1991.

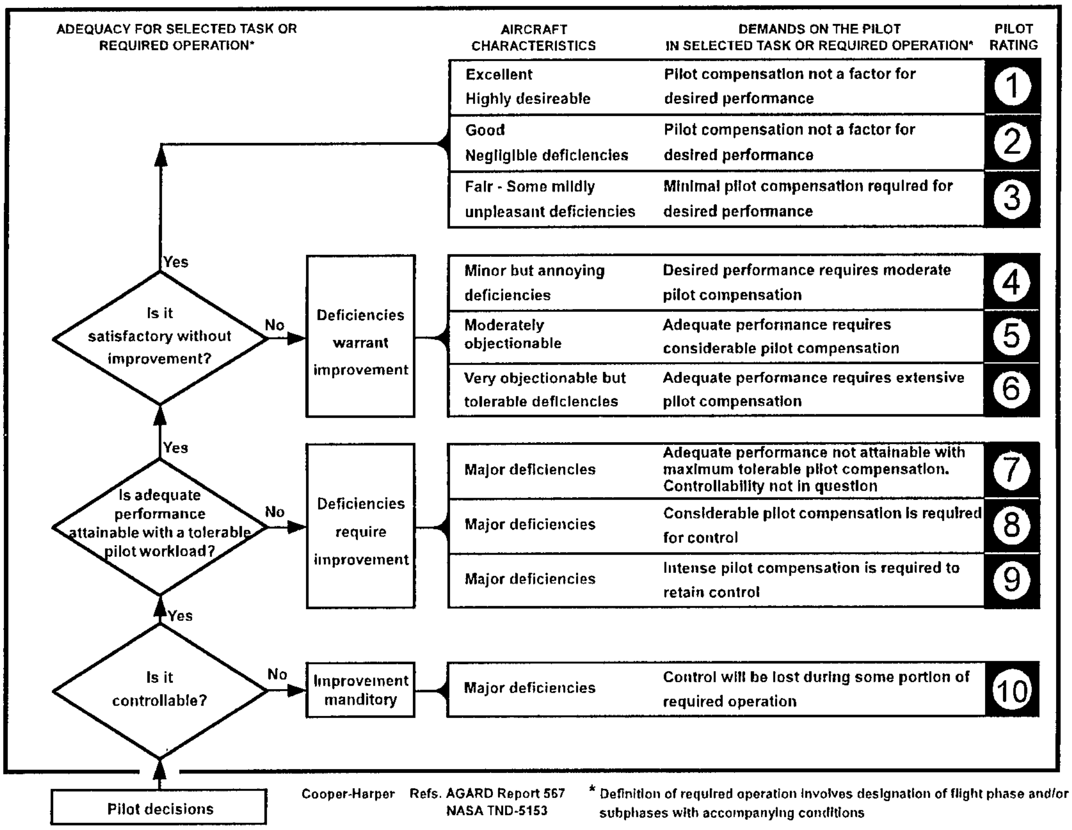


Figure 1 Cooper Harper handling qualities rating scale

Symptoms	Discomfort	Rating
No unpleasant sensations or discomfort experienced.	**None**	**0**
Unpleasant sensations detectable but easily disregarded. Feels slightly uncomfortable.	**Mild**	**3**
Unpleasant sensations moderately intrusive. Feels uncomfortable.	**Moderate**	**6**
Unpleasant sensations very intrusive. Feels very uncomfortable.	**Severe**	**9**
Extremely unpleasant sensations. Discomfort intolerable.	**Unacceptable**	**12**

Figure 2 Discomfort rating scale

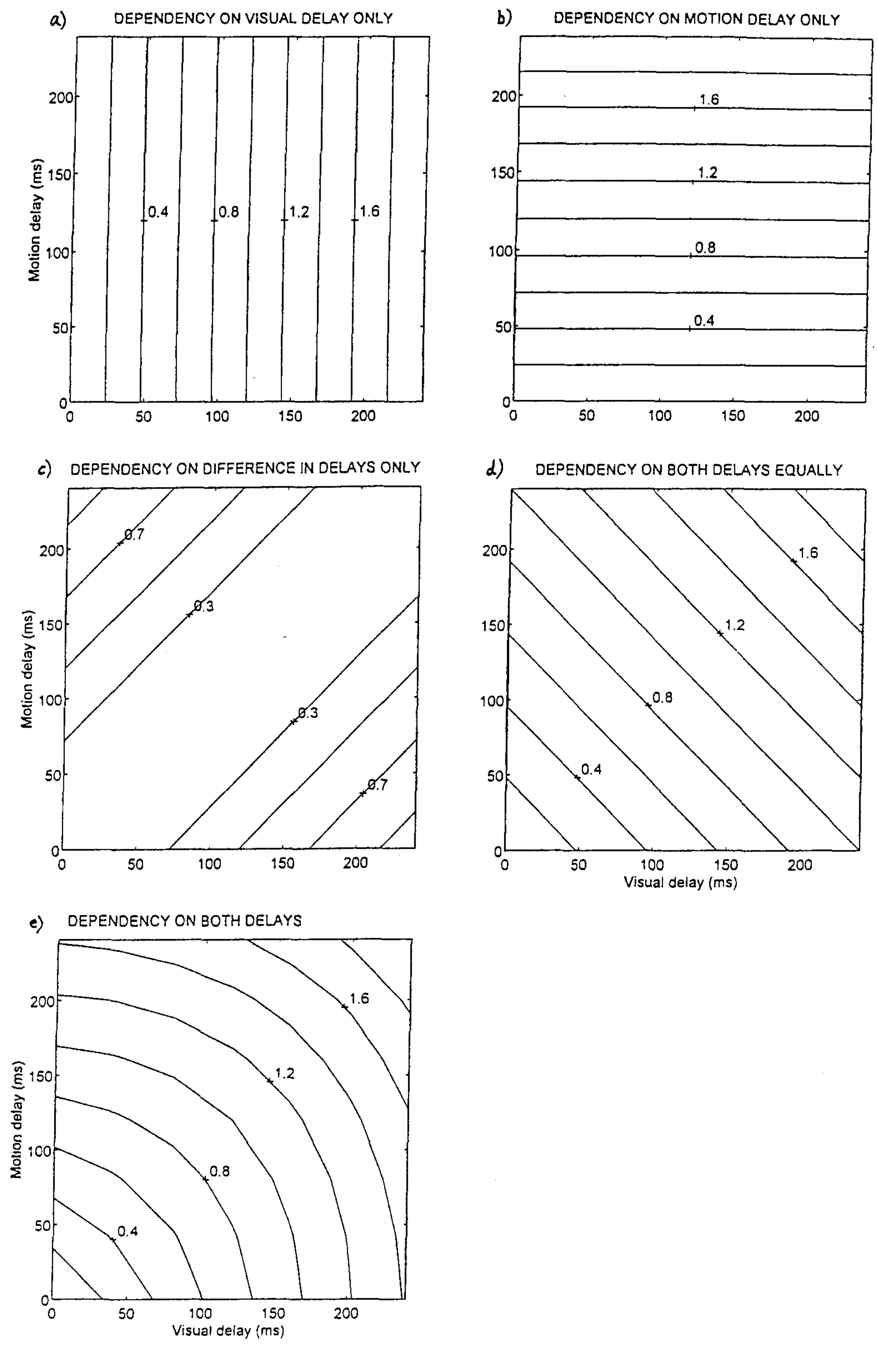


Figure 3 **Examples of function contours assuming a variety of dependencies on motion and visual delays**

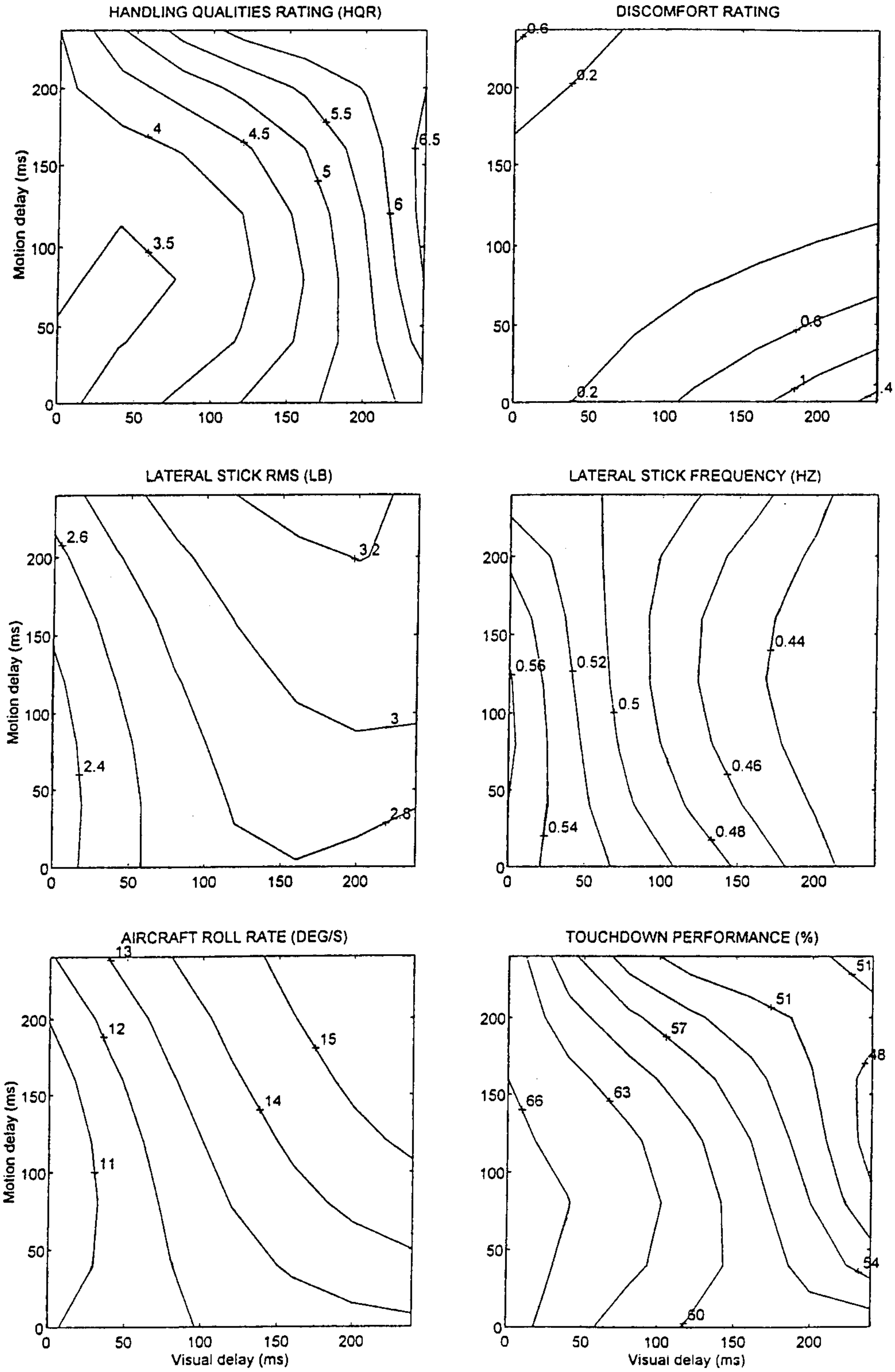


Figure 4 The effects of motion and visual delays in a full motion cueing environment

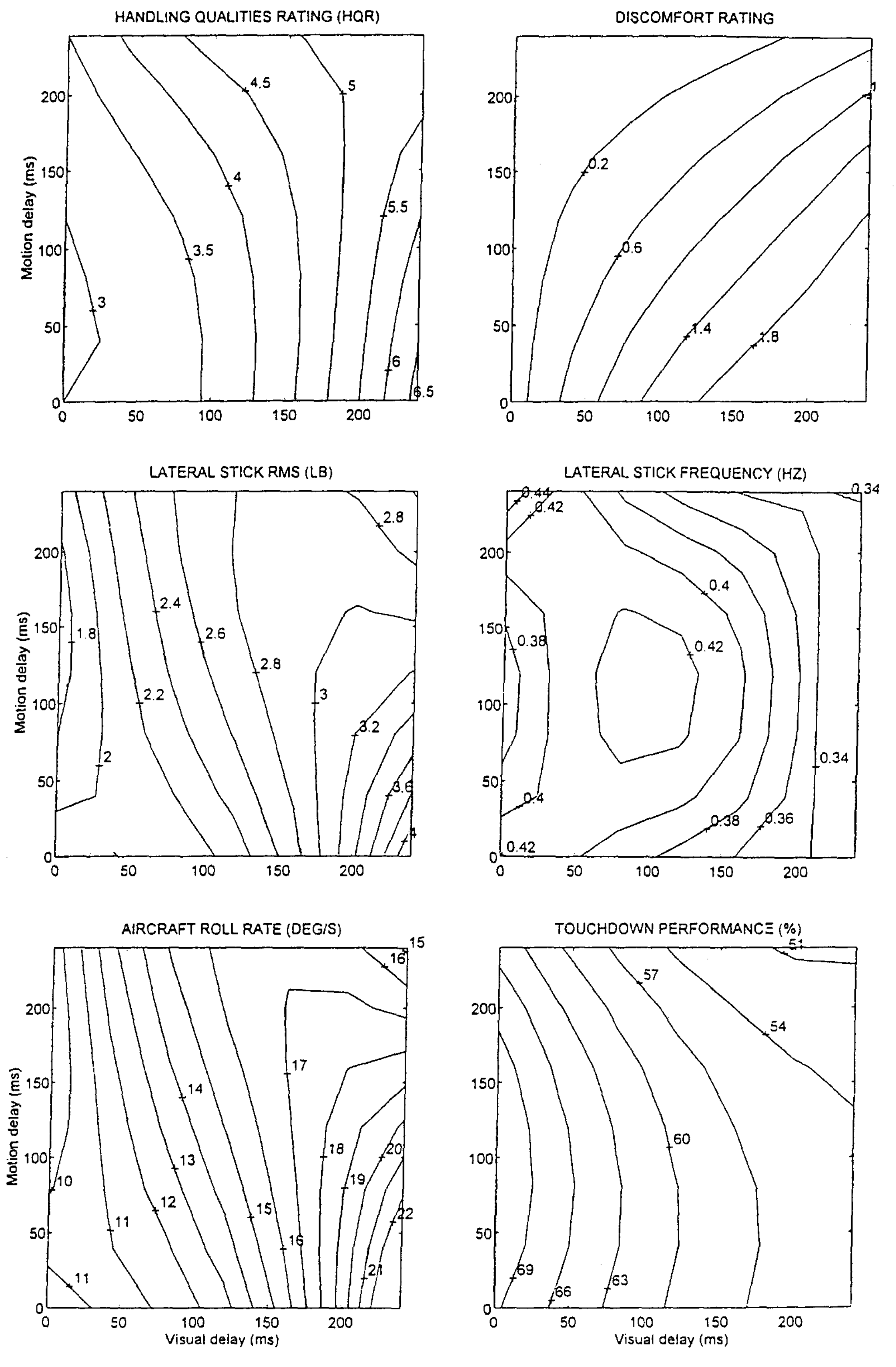


Figure 5 **The effects of motion and visual delays in a conventional platform motion cueing environment**

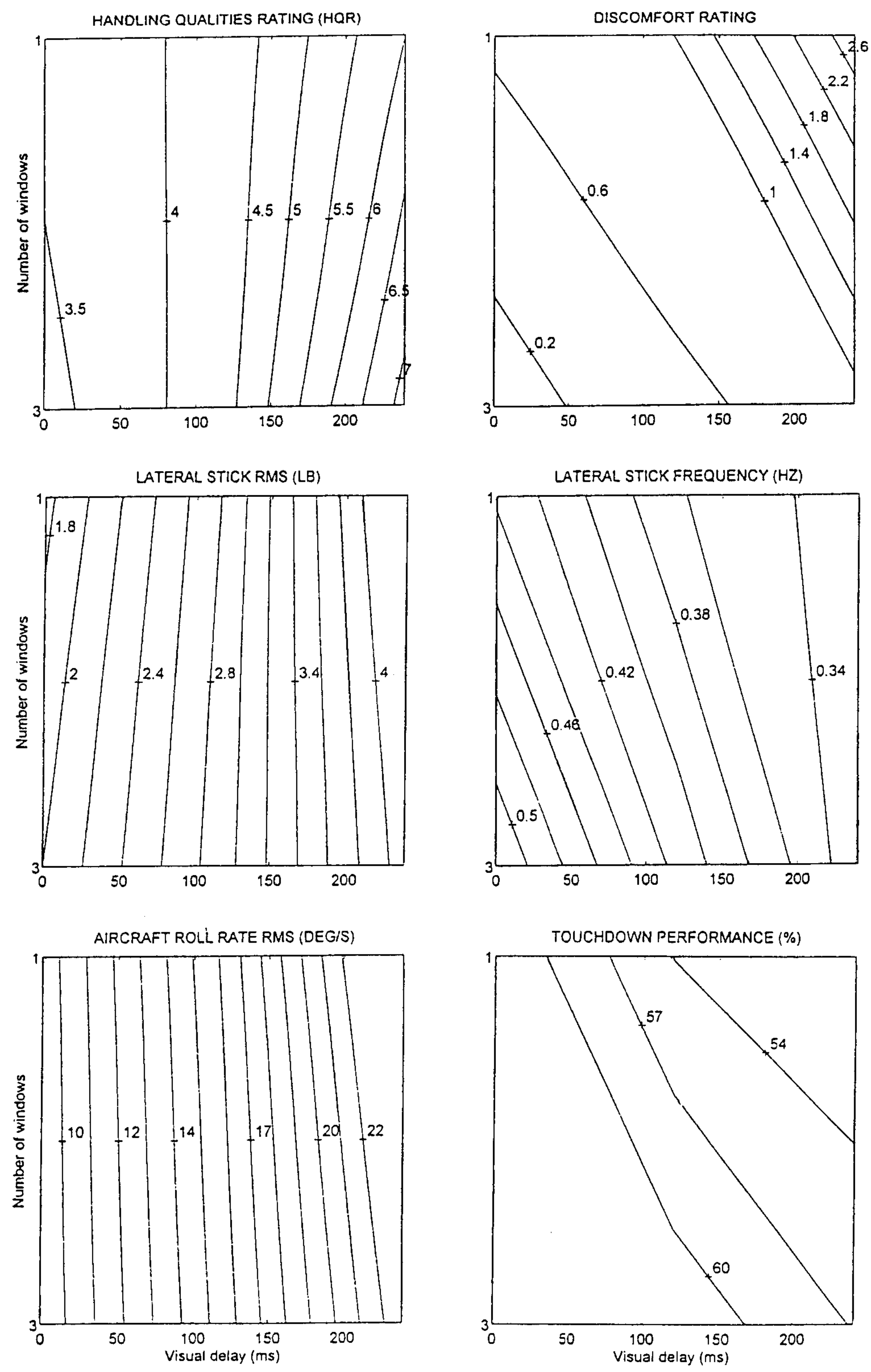


Figure 6 The effects of visual field of view and delay in a fixed-base cueing environment

FEEDING HUNGRY PROCESSORS: REAL-TIME I/O DEMANDS OF HIGH-PERFORMANCE MULTIPROCESSING COMPUTERS

Bruce H. Johnson
Silicon Graphics Computer Systems
Houston, Texas

ABSTRACT

It has been documented that microprocessor performance doubles about every 21 months. Much is published and reported on the technology that delivers this impressive computing power. Much less is said, however, about the unique Input/Output (I/O) demands that are presented when using these microprocessors in high-performance, real-time, multiprocessing environments. Raw computing power is seldom questioned anymore. Of more concern today is the ability of a computer system to deliver data to and from these high-performance processors.

For example, it is not difficult to select a computing engine that is capable of performing the computations necessary to drive a Full Flight Simulator (FFS) or a Weapons Systems Trainer (WST). It is, however, a significantly greater challenge to determine how the simulation I/O can be performed so as to eliminate bottlenecks and latencies. The training value of a simulator can easily be lowered by the stepping or jumping of an instrument, visual system, or motion base that is due to the inability of the I/O to keep up with the processors.

This paper will explore some of the technology available that can be used to "feed" today's high-performance, real-time, multiprocessing systems. Both advances in hardware and software will be discussed, advances that give developers the tools they need to deliver I/O to and from a simulator with determinism and realism.

ABOUT THE AUTHOR

Mr. Bruce Johnson is a Systems Engineer for Silicon Graphics Computer Systems and consults to the real-time, simulation industry. He has spent the last twelve years working for both simulation/training contractors and computer system manufacturers. He holds a Bachelor of Science Degree in Engineering and Computer Science from the University of Florida.

FEEDING HUNGRY PROCESSORS: REAL-TIME I/O DEMANDS OF HIGH-PERFORMANCE MULTIPROCESSING COMPUTERS

Bruce H. Johnson
Silicon Graphics Computer Systems
Houston Texas

INTRODUCTION

There is no doubt that I/O issues are important to the simulation community. Virtually all simulation systems have some sort of specific I/O requirements that must be met. In some cases, the industry has even defined specific I/O standards for simulators. For example, the Federal Aviation Authority (FAA) requires that FAA certified simulators have a transport delay of less than 150 milliseconds (FAA Advisory Circular AC 120-40B). Though not as easily quantified as transport delay, nearly every simulator specification requires that all components be free of jumping and stepping. A simulator does not have to be directly involved with pilot training in order to have specific I/O requirements. There are also some very specific transport delay requirements identified in the Communication Architecture for Distributed Interactive Simulation (CADIS) document.

How data is moved between the simulator and the CPU (or CPUs) can make a big difference in the amount of effort it will take to meet these types of requirements. While the heart of a real-time, host computer system may still be the CPU (or CPUs), the I/O interconnect and how well it is utilized will often determine the level of realism and fidelity of a given simulation.

The computer systems of today differ dramatically from those offered only a few years ago. Along with the changes in features and capabilities, there comes a need to change the way that much of the system design is being done. This is particularly true in terms of designing and optimizing the flow of data from a simulator, to the CPUs, and then back again. Practices that provided efficient use of resources a few years ago could potentially "bottleneck" today's high-performance systems that rely heavily upon data buffering and pipelining.

THE FIRST STEP: PASSING DATA TO/FROM MEMORY

When considering the I/O of a high-performance computer system, the first area to explore is the path from processors to memory. Certainly not unique to the real-time industry, all computing applications are concerned with providing the widest, shortest path from CPUs to memory. Memory subsystems of modern computers are generally made up of Dynamic Random Access Memory (DRAM). While it was stated previously that CPU performance has been doubling every 21 months, DRAM access times have decreased only by a factor of about one third over the past decade. This difference in performance change is shown in Figure 1.

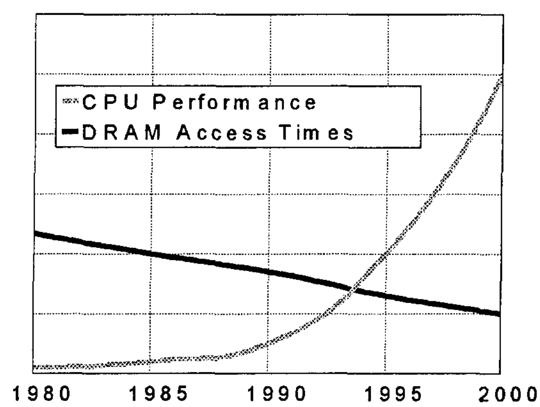


Figure 1

Amdahl's law says that the performance of a computer system as a whole will increase as a result of better performance in a component only to the extent that the improved component may be used. For example, if the speed of a CPU is dramatically increased, but the memory speeds are kept relatively constant, the CPU will spend increasing numbers of clock cycles waiting for memory to respond. Computer designers must clearly provide some techniques that permit faster paths to memory in order to keep the CPUs of the system well fed.

One solution to this problem is in the use of higher speed Static Random Access Memory (SRAM). Ideally, to maximize the bandwidth between CPU and memory (and also provide more deterministic memory access times), all memory subsystems would consist entirely of SRAM memory. Indeed some computer systems offer (or have offered) large SRAM memory subsystems as key components of their design (e.g., Encore RSX and CRAY C-90). The downside to SRAM is in its additional cost when compared to that of DRAM. As can be seen from the following equations, in order to equip an average size simulator host computer with SRAM as opposed to DRAM would increase its price approximately 100,000 dollars.

200$ / MegaByte (MB) * 64MB DRAM = $ 12,800
2000$ / MegaByte (MB) * 64MB SRAM = $128,000

Realtime developers and engineers demand high-performance and determinism, but delivering a system with entirely SRAM memory is certainly cost prohibitive to most.

Overview of Caching

A more cost effective solution to the problem lies in the use of caches. Caches are SRAM-based subsets of the lower cost DRAM that are provided on a system so that the CPUs have faster access to data. Caches are generally implemented with either a "write-through" or "write-back" strategy. Write-through caches will write modifications to their memory immediately back to main memory. Write-back caches will wait to write modified memory back to main memory until the cache line is needed to hold other data. For almost all applications, write back caches provide better performance by reducing the amount of traffic between the cache and main memory.

When cache data is read in, it is normally done using a "cache line" of data. Reading data in cache lines takes advantage of the fact that the next piece of data that a processor wants is very likely to be near the one that was just accessed. This is why, when designing I/O transfers, an engineer should always think in terms of long bursts of data (i.e., a cache line size) since it is likely that this is how data is being moved internally through the system.

All popular RISC microprocessors today incorporate small on-chip caches that generally range from 8 - 128 KB in size. These on-chip caches, or primary caches, are most often supplemented with larger secondary caches that reside external to the CPU itself. The size of secondary caches varies even more, secondary caches exist that are as large as 4MB per CPU. To emphasize the importance of caches in application performance, Table 1 is presented to show access cycles of caches versus that of main memory (numbers are representative only and do not necessarily reflect any particular computer system):

read cache hit primary cache	1 cycle
read cache hit secondary cache	4-11 cycles
read miss (access main memory)	100-160 cycles

Table 1

Caching and Multiprocessing

The use of cache in computer systems is not a new design concept, but integrating of cache with multiprocessor computer systems is. The challenge for a multiprocessing system when using caches is to ensure that each CPU has a consistent view of the system memory. When using caches, it is certainly not uncommon for the same data to be kept in multiple locations and this data must be efficiently synchronized and exchanged. A typical solution to this problem is through the addition of bus snooping to the system architecture. Bus snooping means that each interface to the system bus (i.e., CPU cards, memory cards, I/O interfaces, etc.) monitors, or snoops, the bus traffic. Upon a read request, the interface with the latest copy of the data responds to the read. The techniques implemented by multiprocessing systems to optimize cache coherency can make a big difference in overall system performance.

Some multiprocessing systems implement a duplicate set of cache "tags" in order to minimize contention between processors and the bus snooping mechanism. Cache tags are used in a system to identify the addresses for the data that reside in cache. They are used by both the CPUs and bus snooping mechanism in a multiprocessor system. The CPUs use cache tags to index into the cache and pull data when there is a cache hit. The bus snooping mechanism utilizes cache tags when monitoring the addresses on the bus. When snooping and CPU access to cache occur simultaneously, CPU cycles can be lost due to contention for tags. Duplicate cache tags can greatly reduce the contention between CPU and the bus snooping mechanism.

Another cache coherency technique used in some architectures is the three-party transaction. A three-party transaction is implemented as follows. Whenever a read request is satisfied by data from a second processor's cache, the main memory interface "accepts" the read as if it were a write. For example, if processor A accesses memory for which the only valid copy is contained in processor B's cache, the block of memory will be simultaneously written back to main memory and transferred to processor A's cache.

Memory Interleaving

Another technique used for maximizing the bandwidth to memory is memory interleaving. Memory interleaving is a way of organizing the DRAM memory of a system into "leaves" that are capable of independently processing a memory request for a processor. Each memory leaf can be thought of as an independent path from CPUs to memory. By interleaving memory, the negative affect of DRAM access latency (when compared to CPU clock rate) is hidden by overlapped memory accesses in multiple memory leaves.

Memory interleaving is especially advantageous in multiprocessor systems. Most operating systems ensure proper distribution of sequential memory accesses by a single process. However, this is normally not done for multiple processes executing on multiple processors. Therefore, to minimize the probability of two successive memory accesses to the same leaf, memory interleaving is provided. As the number of processors increases, the number of memory leaves should have the capability of being increased as well.

THE NEXT STEP: PASSING DATA TO/FROM THE I/O BUSES

As with advances in the access times of memory, growth in the performance of I/O buses has not kept up with growth in CPU performance. Figure 2 and Table 2 compare the growth in the performance of industry standard I/O buses with that of RISC microprocessor performance (years indicates years of use, not introduction).

Years	Popular I/O Bus	Bandwidth
< 1980	MultiBus-1	10 MB/sec
1980s	SELbus	26.67 MB/sec
1990s	VMEbus	40-65 MB/sec
2000s	Futurebus+	100 MB/sec

Table 2

Table 2

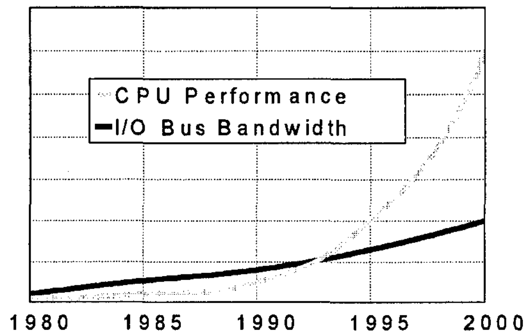


Figure 2

Not only has I/O bus performance growth lagged CPU performance growth, but the data path from CPU to I/O bus in today's systems is generally longer than what it used to be. Data often must be scaled from the large internal buses of RISC processors to narrower I/O buses such as SCSI and VME.

Figure 3 is a simplified example of how the buses of a high-performance computer system are often scaled down from larger system buses to smaller I/O buses.

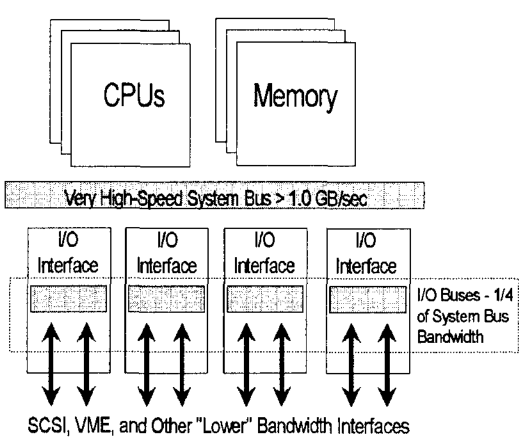


Figure 3

In a similar fashion, the bus bandwidth of the system bus is generally much higher than that of I/O and peripheral buses. This presents system designers with the unique challenge of trying to maximize the throughput of a lower performance bus (i.e., SCSI or VME), while minimizing the negative effect that the slower data transfers will have on the high-speed system bus. Some features that are used to accomplish this are I/O caches, data buffering, and the pipelining of bus grant and requests on the I/O bus.

Maximizing the Performance of a VMEbus

Whether incorporating a single board interface or a complete I/O subsystem, the VMEbus has become a de-facto standard in the simulation and training industry. While most real-time computer systems are designed with an integral VMEbus, the performance features and integration capabilities of the bus can often vary greatly between manufacturers.

In theory, boards that comply with the IEEE-1014 VMEbus Specification should work together on the same bus. In practice, however, every design engineer knows that VME board integration can be a difficult task. At least part of the problem can be attributed to the format of the VMEbus Specification as it leaves room for interpretation in many areas. As a result, boards can be functionally correct yet employ marginal design practices that are exposed with increased bus activity and contention. In addition, a myriad of software issues can increase the complexity of VME board integration as well.

VME board selection is crucial to optimizing the bandwidth of the VMEbus. The bandwidth of the VMEbus in a given application will most likely be determined by the boards on the bus, not the host computer controlling the bus. Boards that support block transfers, Direct Memory Access (DMA) transfers, and 64-bit data transfers are going to be able to pass data at much faster rates than those that support only Programmed I/O (PIO). Table 3 shows an example of the differences in throughput that can be expected between data transferred via PIO versus that of DMA (while data is representative in nature it is based on actual testing results).

Transfer Type	PIO Bandwidth	DMA Bandwidth
8-bit(D8) Read	.5 MB/sec	3.0MB/sec
8-bit(D8) Write	.75MB/sec	3.75MB/sec
32-bit(D32) Read	2.0MB/sec	12MB/sec
32-bit(D32) Write	3.0MB/sec	15MB/sec

Table 3

Some of the Features Available for VME I/O

One feature that several real-time computer systems are offering is the capability to support multiple VME buses from a single computer. In fact, systems are available today that offer up to five separate VME buses per system. Not only does this dramatically increase the I/O bandwidth of a system but it also adds a great deal of flexibility to the I/O design. For example, it is often a wise practice to isolate slower VME boards (i.e., those that are accessed via programmed I/O) from boards that are performing faster DMA block transfers.

Many vendors now also provide support for 64 bit data transfers on the VMEbus per Revision D of the VME Specification. 64-bit transfers can effectively double the throughput capabilities of a VMEbus. In order to implement 64-bit transfers, both the host and the target (VME board) must provide D64 support. It appears that more and more VME board manufacturers are adding this capability to their products.

For VME boards that do not have DMA capabilities, some host computer vendors provide a DMA master capability on their VMEbus interfaces. Commonly referred to as a DMA engine, this DMA master capability enables boards that lack support for DMA to increase their performance by using the DMA capabilities of the VME interface. Using the DMA engine facility, applications have been shown to deliver higher performance even when using transfer sizes as small as 128 bytes. Figure 4 shows relative performance differences between I/O transfers using PIO compared with those of a typical DMA engine.

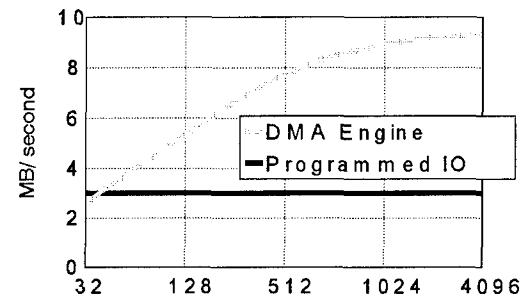


Figure 4

Perhaps unique to real-time systems is a feature that gives the capability to route VMEbus interrupt levels to individual CPUs. In this way, real-time tasks can be isolated to execute in response to interrupts mapped to a specific CPU - thereby reducing interrupt latencies incurred by the processing of non-realtime interrupts on the CPU. Additionally, for systems that operate under a UNIX-based OS, it is virtually imperative that the clock scheduling interrupt be removed from an isolated processor to ensure determinism in responding to high priority interrupts.

Calculating Available Bus Bandwidth

A very important metric in determining the I/O capabilities of a computer system is the available bus bandwidth. Available bus bandwidth refers to the amount of data that an I/O backplane can transfer at any one time. Often vendors quote an I/O bus bandwidth that reflects only the performance of a single type of operation - invariably, the fastest. In almost all simulation applications, however, the I/O bus incorporates several different types and sizes of transfers on the bus.

Available bus bandwidth varies greatly with each application and is dependent upon both the host and the I/O boards. The following example demonstrates this. A given simulation application has the following I/O requirements every frame of its simulation:

5 KB	read from reflective memory
30 KB	written to reflective memory
100 KB	read from DMA interface device
100 KB	write to VME-based SCSI disk for record/playback

For realistic data presentation the I/O is performed at 60 Hertz (Hz):

5 KB * 60 =	0.3 MB
30 KB * 60 =	1.8 MB
100 KB * 60 =	6.0 MB
100 KB * 60 =	6.0 MB
Total =	**14.1 MB per second**

Only 14.1 MB/sec need to be sustained, which is well within the vendor quoted bus bandwidth of 25 MB/second. Yet taking a closer look at the type of I/O that is actually being performed reveals a real I/O bottleneck. For purposes of this discussion, assume the computer system is capable of delivering the following bandwidths with various operations on the VMEbus:

D32 PIO Non-block write transfers	4 MB/sec
D32 PIO Non-block write transfers	1 MB/sec
D32 DMA Block read transfers	30 MB/sec
D64 DMA Block write transfers	55 MB/sec

Applying these numbers to our previously generated throughput requirements yields a problem:

1.8 MB @ 4 MB/sec =	45%
0.3 MB @ 1 MB/sec =	30%
6.0 MB @ 30 MB/sec =	20%
6.0 MB @ 55 MB/sec =	11%
	106% utilization

A solution to this engineer's problem could lie in the availability of a VME DMA master capability on the host computer which could be used with VME cards that support only PIO. If provided, this DMA "engine" should be able to deliver the following bandwidth for writes to the VMEbus:

DMA Engine D32 Nonblock writes 9 MB/sec

Plugging this into the equations, we now have a different result:

1.8 MB @ 9 MB/sec =	20%
0.3 MB @ 1 MB/sec =	30%
6.0 MB @ 30 MB/sec =	20%
6.0 MB @ 55 MB/sec =	22%
	92% utilization

These numbers are given only as an example, yet they underscore the importance of understanding the various factors that influence available bus bandwidth.

Using SCSI for Realtime

SCSI controllers are inexpensive DMA devices that are generally optimized to make maximum use of a system's resources. When using 16-bit SCSI, useful bandwidths of up to 14 MB/sec can be achieved on each SCSI bus. Factoring this into an average price for bus expansion and comparing it to VME yields the insightful results of Table 4.

Device to Add	Price	Bandwidth	$/MB
SCSI bus	< $1000	14 MB/sec	71
VMEbus	> $ 10000	50 MB/sec	200

Table 4

While SCSI may still present a non-traditional approach to interfacing and controlling real-time hardware, its price/performance advantage certainly cannot be ignored. Already, commercial interface manufacturers are realizing the advantages of SCSI by introducing SCSI-based I/O devices such as terminal servers. Real-time SCSI interfaces should be considered as a cost effective solution for virtually any future I/O requirement.

MULTIPROCESSING AND I/O

While very powerful simulators and simulation systems are being developed today that are driven by single-processor computer systems, multiprocessor systems are becoming more and more the norm. A few of the reasons why multiprocessor systems are advantageous for simulation applications are identified in the following paragraphs:

- Many simulators and simulated systems have increased in complexity of late and require additional system resources to process the load. Development of simulators for such devices as the Air Force's F-22 and NASA's Space Station Freedom are reaching new limits in processing power demands.

- Recent trends show an increase in the incorporation of special purpose processing at the host computer level. Image generation graphics, motion-base flight models, control loading forces, DIS packet processing, etc., are now all capable of being processed by host computer processors. This design permits better data localization and simpler process synchronization than do designs that incorporate loosely-coupled processors.

- Multiprocessing also permits isolation of processors to perform specific real-time tasks. Processor isolation means that all ancillary processes to the real-time process (i.e., system daemons, graphics processing, etc.) can be forced away from real-time CPUs, thereby achieving improved realtime response and determinism.

Symmetric Multiprocessing and I/O

A Symmetric Multiprocessing (SMP) system is a balanced computational system where multiple processors equally share the resources of the system in order to process a given workload. The sharing of resources is generally done via a shared system bus that connects the CPUs, memory, and I/O of the system. In a true SMP system, ALL processors of a system have equal access to ALL resources of the system including memory, I/O, and the Operating System (OS) itself.

This simple fact is a key to maximizing the advantage of an SMP system. Not all multiprocessing systems are fully symmetric and, therefore, leave a potential for different performance results based on varying factors of asymmetry. For example, a multiprocessor system may have memory areas that favor certain processors such that there can be dramatic differences in performance based on which memory is being utilized by which processor.

Likewise, I/O interfaces that reside on the CPU boards of a system often provide fast access to peripheral devices from only a subset of the available CPUs. The remaining CPUs of the system are likely to either have much slower access to the interfaces or no access at all. In general, an SMP system will be more predictable in nature if resources of the system are able to be evenly distributed to the available CPUs of the system.

DISPELLING SOME COMMON MISCONCEPTIONS ABOUT REALTIME SIMULATION I/O

"Realtime simulation applications have poor cache hit ratios"

While this statement was largely true for computer systems that were being designed and manufactured only a few years ago, it is far less true today. It is rather difficult to generalize about the characteristics of all simulation code, but for purposes of this discussion a few assumptions will be made. Simulation code, as compared to code of a purely scientific application or that of a DataBase Management System (DBMS), has a greater percentage of branch instructions in its instruction stream and a larger percentage of scalar data located randomly in memory. Because of these characteristics, simulation code does not take advantage of the cache as well as do other applications.

Yet the reason that simulation applications are caching better today is due to the simple fact that the size of caches has dramatically increased over the past few years. Table 5 shows examples of how cache sizes of simulation host computers have increased over the past ten years.

Year	Example Cache per Processor
1980	32 KB
1985	up to 128 KB
1990	up to 256 KB
1994	up to 4MB

Table 5

So substantially have cache sizes increased, that many industry "cache busting" benchmarks no longer bust caches. By the time this paper is published, there will be high-performance computer systems available with up to 4 MBytes of cache per processor. With cache sizes this large, simulation code now caches better than ever.

"Multiprocessing systems will saturate their system bus any time more than a few processors are added to the system"

As with the previous misconception, this statement is much less true today than it was a few years back. The main reason for this is in the dramatic increase in system bus bandwidth in recent years. Table 6 shows examples of changes in system bus bandwidth over the past ten years.

Year	SystemBus Bandwidth
1985	26.67 MB/sec
1990	100 MB/sec
1994	1200 MB/sec

Table 6

"I/O cycles can easily "starve" the system bus away from the CPUs"

This is a misconception that often is brought about by a lack of understanding of how data is actually transferred throughout a high performance computer system. Here is a classic example of how this misconception is formulated. A systems engineer is trying to understand the way that DMA transfers occur in his computer system in order to estimate what resources are being consumed:

1. His computer system has a sustained system bus (or memory bus) bandwidth of 1.2 Gigabytes/second.
2. His application requires a SCSI device to perform 2 KB DMA transfers into main memory at a rate of 5 MB/second.
3. He concludes that once the data transfer phase begins, his transfers will require about 400 microseconds (μs) to complete and will tie up the system bus during that period of time.

Taking a closer look at how data actually flows through this system, however, reveals quite a difference. The basic unit of transfer on this system bus is actually 128 bytes (the size of a full cache line of data) and the bus transfers each 128 byte block in 100 nanoseconds (ns). Since the interface to the SCSI bus contains enough buffering and data funnel operations to pack all operations and keep the system bus from being used inefficiently, the system bus will be used for only 16 100ns transfers. This equates to 1.6 μs on the system bus - hundreds of times less than what was originally expected.

IMPORTANCE OF UNDERSTANDING HARDWARE ISSUES AT SOFTWARE LEVEL

As the complexity of simulators and simulation systems increases, there are more and more attempts to minimize the complexity of the system to the average software developer. Many higher level languages provide constructs that encourage abstraction in an effort to simplify the amount of detail that must be understood by each programmer. For the most part, these concepts are desirable and should be encouraged. Yet this can lead to some dramatic performance implications if details of how I/O is handled by the system are not well understood.

Programmed I/O Pitfalls

By using standard system calls of virtually any OS, one may "map" VME address space to the virtual memory of a user process. This permits a user to easily read/write to the memory and registers of a VME board as if using global variables of their program. A user process then may program the VME card to perform different functions and/or send and receive data through Programmed I/O (PIO) reads/writes. Reflective memory boards are often integrated this way and provide a simple way for separate computer systems to share memory.

There are, however, several potential problems with implementing PIO that should be considered when evaluating for potential I/O bottlenecks. First of all, from a systems point of view, PIO is a very inefficient way to move data around. PIO operations cannot be pipelined and the entire I/O path from the VMEbus to the CPU remains "tied up" during the PIO operation. In addition, tests have shown that PIO utilizes 100% of a CPU's resources during large I/O transfers - leaving it unusable to other processes. This contrasts with DMA transfers which can be extensively pipelined through the use of data prefetching, freeing up the CPUs considerably.

What can also happen when using PIO is that developers may begin to treat the memory that resides on the VME as they would the normal virtual memory of their process. Simulation variables are often defined and manipulated on the VME-mapped memory such that hundreds of VME accesses are occurring with every pass through their programming model. In many cases the software developer is not even aware that VME memory is being accessed since often the address mapping occurs external to their process or program. The results of this can be disastrous. A read or write to VME memory can take hundreds of times longer than that of normal memory access.

Pipelining, Caching, Etc.

Another potential area for severe performance degradation that is often overlooked by the average software developer is the caching characteristics of a data area. As an example, the software design may utilize a large shared memory area (or datapool) for the sharing of data between processes. Then a portion of this area may be accessed from a DMA master that resides on the VMEbus. Since certain system architectures will mark entire DMA buffers uncacheable, developers must be careful that the entire shared memory area is not also marked uncacheable. To maximize the use of cacheable shared memory partitions, it is often advisable to split up normal data areas from those that will be used for I/O (i.e., multiple datapools or memory partitions).

CONCLUSION

Computer systems today offer incredulous computing power from the desktop to the supercomputer. Yet even the most powerful of computing engines cannot guarantee that a simulation be free of excessive I/O latencies and "glitches". Effectively harnessing the available computing power in the simulation and training community requires both an understanding of computer system architecture and the application itself.

This paper has presented some details about the real-time, I/O capabilities of today's high-performance, multiprocessing computer systems. It has also presented some design hints that can be used to minimize I/O problems that invariably crop up during simulator design (or even long after simulator delivery). While today's processors are hungrier than ever, understanding how to keep them well fed can ultimately help the industry to meet the challenges of the 1990s and beyond.

Technical Expectations for a Full Scale Domain Engineering Demonstration Project

Lynn D. Stuckey, Jr. David C. Gross
Boeing Defense and Space Group

Greg D. Pryor
Naval Air Warfare Center Training Systems Division

ABSTRACT

STARS (Software Technology for Adaptable, Reliable Systems) is a long-term ARPA project aimed at advancing the management, quality, adaptability, and reliability of DoD software intensive systems Over the years, the STARS project has gradually focused on enabling a paradigm shift of DoD software practices to *megaprogramming.* The central megaprogramming concept is a process-driven, two-life-cycle approach to software development. One life-cycle spans the creation and enrichment of an organization's capabilities for a family of related products, or *domain engineering.* The other life-cycle spans the construction and delivery of individual products, or *instances* from the domain. This approach may provide substantial opportunity for *leveraged reuse*, that is, planned use of adapted software components in multiple products. Much of the effort to date has been for developing tools and processes that support megaprogramming. The STARS project is now in a transition and demonstration phase. One of the demonstration projects is in the domain of simulator-based training, specifically the U.S. Navy's domain of Air Vehicle Training Systems. If megaprogramming proves useful in this domain, it promises dramatic increases in productivity along with corresponding reductions in the cost of building simulations.

Previous experience reports have focused on pilot efforts in domain engineering for sub-domains of the Air Vehicle Training Systems (AVTS) domain such as environment and navigation simulation. These pilot efforts have demonstrated that the processes and tools are sufficiently mature for full scale domain engineering for AVTS -- which the demonstration project is proceeding to do. This paper summarizes the lessons learned from the pilot efforts and looks ahead to the technical challenges and opportunities we anticipate in the full scale demonstration. Expected technical challenges and opportunities include:

(a) Integrating many sub-domains,
(b) Relating to non-domain engineered models,
(c) Integrating dramatically larger staffs,
(d) Relating to a real product acquisition project,
(e) Controlling adaptation, and
(f) Leveraging extra-domain assets.

ABOUT THE AUTHORS

Lynn D. Stuckey, Jr. is a software systems engineer with the Missiles & Space Division of the Boeing Defense & Space Group. He has been responsible for software design, code, test, and integration on several Boeing simulation projects. He is currently involved in research and development activities dealing with software reuse in the domain of air vehicle training systems. Mr. Stuckey holds a Bachelor of Science degree in Electrical Engineering and a Master of Systems Engineering from the University of Alabama, in Huntsville. His thesis presents a systems engineering approach to software development. Mr. Stuckey is a doctoral student at the University of Central Florida. He can be reached as stuckey@plato.ds.boeing.com

David C. Gross is a software systems engineer with the Missiles & Space Division of the Boeing Defense & Space Group. He has worked in all life-cycle phases of simulation and training systems from requirements development through delivery. He is currently involved in applied research related to megaprogramming in the domain of training simulator systems. Mr. Gross holds a Bachelor of Science in Computer Science/Engineering from Auburn University and a Master of Operations Research at the University of Alabama at Huntsville. His thesis compares the utility of high level languages such as C++ and Ada for simulation. Mr. Gross is a doctoral student at the University of Central Florida. He can be reached as gross@plato.ds.boeing.com

Greg D. Pryor is a computer engineer with the Naval Air Warfare Center Training Systems Division (NAWCTSD). Prior to 1994, he worked at the Navy's Manned Flight Simulator. He was lead aerodynamics engineer for the MFS AH-1W Aircrew Procedures Trainer. His current assignments for NAWCTSD include the Navy/ARPA STARS demonstration project in the Air Vehicle Training Systems domain, and the AH-1W Cobra simulator. Mr. Pryor holds a Bachelor of Science in Aerospace Engineering from North Caroline State University, and a graduate of the United States Naval Test Pilots School. He can be reached as pryorg@ntsc.navy.mil

Technical Expectations for a Full Scale Domain Engineering Demonstration Project

Lynn D. Stuckey, Jr. David C. Gross
Boeing Defense and Space Group

Greg D. Pryor
Naval Air Warfare Center Training Systems Division

INTRODUCTION

The software community's experiences thus far in creating significant amounts of reusable software have seemed to be more wild goose chases than productive endeavors. Reuse projects to date seem to end up with endless lines of code that sit in a library and are never reused. The return on investment has rarely (if ever) justified the effort. Nevertheless, software reusability remains a hotly pursued goal for most organizations. The reasons are simple -- reusable software promises simultaneous progress toward the twin goals of any profit-oriented venture: (a) reduced cost of production, and (b) increased quality. The Software Technology for Adaptable, Reliable Systems (STARS) project aims at reaching the reusability goal through *megaprogramming*.

How is megaprogramming different than earlier efforts? Most reuse efforts have attempted to reuse existing software, either by adopting high quality software or re-engineering existing software. This is an intrinsically flawed approach. Many researchers have noted that reusability is not an accidental characteristic of software. "Reusability is first and foremost a design issue. If a system is not designed with reusability in mind, component interrelationships will be such that reusability cannot be obtained no matter how rigorously coding or documentation rules are followed." [Ausnit85] However, the recognition that creating reusable software is primarily a design issue is a far cry from defining the process for creating such software. And creating reusable software is only the first problem -- any consideration of the human nature of programmers will quickly lead to the conclusion that if the software is to be used, it must be easy to use. This implies some sort of automated, process-driven CASE tools. Finally, the most significant challenge is overcoming the cultural and business barriers to actually practicing reuse. Megaprogramming purports to address these problems.

BACKGROUND

STARS

In June 1983, the Department of Defense endorsed the DoD Software Initiative which had been proposed to offset growing manpower shortfalls and to improve capabilities for supporting software development and maintenance. This new initiative was comprised of three components. One component was the Ada Joint Program Office (AJPO). The second component was a Federally Funded Research and Development Center (FFRDC) later to be called the Software Engineering Institute (SEI). The third component was a project in software technology, which became STARS.

STARS is sponsored by the Advanced Research Projects Agency (ARPA), and involves three cooperating prime contractors -- Boeing, IBM, Paramax -- and a large number of subcontractors including DU-AL, Incorporated and Enzian, Incorporated. Figure 1 illustrates the STARS Approach, based on megaprogramming. Megaprogramming is a product line approach to the creation and maintenance of software intensive systems. It is characterized by a product line view of the reuse of software life-cycle assets (architecture, components, and processes) and a disciplined, process-driven approach to development and evolution of application systems and the product line. The principle benefit to be derived from megaprogramming is improved predictability and quality in software and system development.

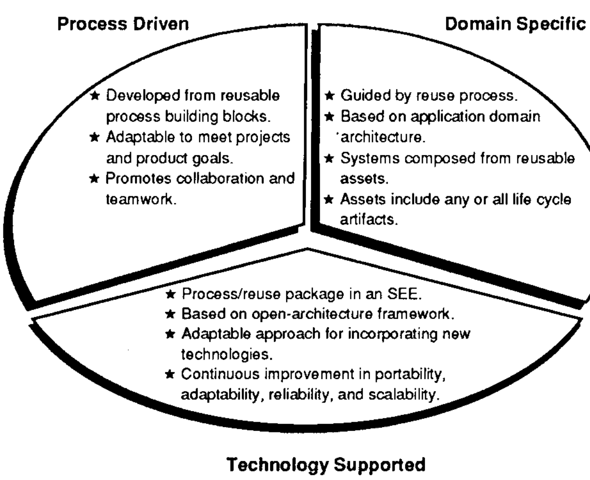


Figure 1: The STARS Approach

Synthesis Process

Engineering knowledge is a standardization of the decision process and its products. If this process captures the commonalities and variabilities between specific systems within a product family, we can create an opportunity to leverage reuse within the product family. In a sense, this is simply formally capturing the engineering knowledge about a family of systems, or domain. The Virginia Center of Excellence (VCOE) has created a process called Synthesis for defining a domain, specifying the products and adaptation for instances for the domain, and implementing designs that leverage whole and adapted components for reuse in new systems. [VCOE93] Figure 2 illustrates the Synthesis process.

Synthesis is a process developed for *leveraged* reuse. Leveraged reuse is one of five approaches to reusing software: (a) ad hoc, (b) opportunistic, (c) integrated, (d) leveraged, and (e) anticipated. Leveraged reuse assumes that a given product is actually a member of a product family, the members of which share some degree of commonality. Synthesis involves the definition, analysis, specification, and

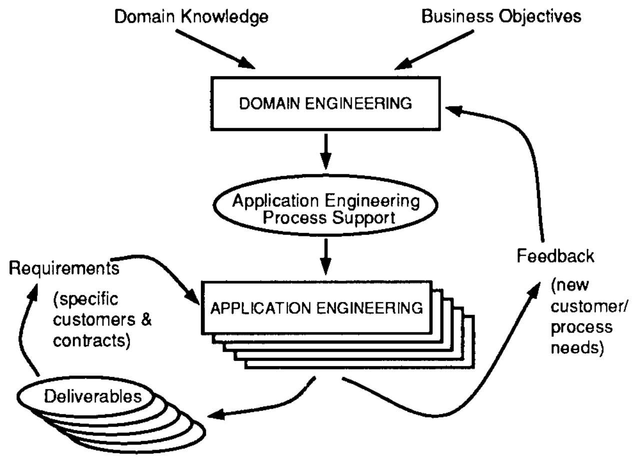


Figure 2: The Synthesis Process

implementation of a domain which encompasses a *viable* product family -- a family which shares sufficient commonality (and predictable variability) to justify an investment in the domain. Individual products are developed as instances of the domain, which reuse common elements of the domain and adapt variable elements using a defined, repeatable process. Synthesis defines the effort associated with creating the assets as *domain* engineering, and the effort in creating a specific product as *application* engineering. The central concept is that domain engineering creates the assets and processes by which application engineering can consider and select the best fitting instance of the domain.

Technology Support

Megaprogramming recognizes that the best reusable software in the world will not be used if it is not convenient to do so. Furthermore, the megaprogramming process encompasses advanced software engineering techniques, which lend themselves to the adoption of CASE tools. The STARS project has responded to these needs by incorporating technology support in the form of integrated Software Engineering Environments (SEE). STARS funding of process engineering and SEE development far in advance of the demonstration projects is evidence of its commitment to its three central concepts: (a) domain specific, (b) process driven, and (c) technology support.

The SEE utilized in this project was developed by the Boeing Company under ARPA funding. The Boeing SEE provides a set of functional services which are integrated in a standard framework, and can be driven by process definitions. Example functional services are those for requirement definition and code development. These functional services support activities in both life-cycles (domain and application engineering). The SEE includes typical computer environment services such as operating systems, editors, and language compilers. It goes beyond this to provide advanced CASE tools. All of these services are supported by commercial off-the-shelf products.

The Boeing SEE integrates the services via a process engine which enforces organizational policies and procedures. For example, the process engine uses defined work breakdown structures and activity precedence networks to schedule and enable software development activities. Naturally, these activities derive from the Synthesis process definition, so the SEE helps an organization "stay on course" by following its desired plan.

The Boeing SEE controls storage and access to adaptable components. Access to components is accomplished via a knowledge-based tool, which uses rules about the domain (identified by domain engineering) to guide selection and adaptation of components.

REVIEW OF PILOT EFFORTS

The STARS project is engaged in three demonstration projects (one per military service, each in cooperation with a prime contractor). The Navy's demonstration project is the only one within the field of simulation and modeling; selected because of its outstanding potential for systematic, long term product improvements. The fundamental concept in the demonstration project is a shift from viewing the construction of software intensive systems as "one at a time", to viewing a specific software system as an instance from a domain. In other words, a software system is really just one product out of a product line. The benefit of this approach is potentially a dramatic increase in the number and kind of reusable components. If their project is successful, the Navy intends to treat training simulators as a domain (product line) -- the Air Vehicle Training Systems (AVTS) domain.

The demonstration project has defined the scope of the AVTS domain, focused thus far on expanding the domain to address the needs of trainers for T-series aircraft. The T-series aircraft consist of the various types of training aircraft in current and future use by the Navy. These aircraft are used in the instruction of student pilots in the various aspects of naval aviation. This demonstration project will develop at least one instance from the domain, a T-34C Flight Instrument Trainer (FIT). If megaprogramming proves useful in this domain, it promises dramatic increases in productivity along with corresponding reductions in the cost of building simulations. This demonstration project has completed its pilot and preparatory efforts, and is entering full scale development. Progress thus far has raised certain expectations about the advantages and disadvantages of the megaprogramming approach for simulation. The demonstration project is organized into four phases as illustrated in Figure 3.

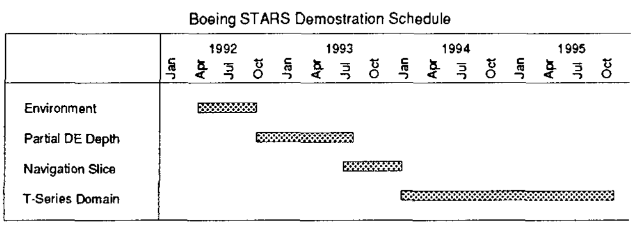


Figure 3: Demonstration Project Schedule

The demonstration project began in 1992 with the application of Synthesis to one aspect of the domain -- the Environment. This effort created a stand alone domain capable of supporting a few instances of the Environment. The next project phase involved application of the preliminary steps in Synthesis across the entire AVTS domain. This effort completed domain definitions, specifications, and preliminary design for the aspects most relevant to T-series aircraft. The third project phase finished the domain design and implementation for a slice of the Navigation sub-domain, and demonstrated the slice in the context of the AVTS architecture. The current phase of the project is to continue the domain engineering process in order to expand the AVTS domain to more completely support T-series aircraft, specifically a T-34C FIT instance. This phase will conclude with the demonstration of a functioning T-34C FIT in October of 1995. The following paragraphs briefly describe each of the pilot efforts along with some pertinent lessons learned.

Environment Segment Pilot

In 1992 Boeing Simulation & Training Systems (S&TS) performed a pilot project using Synthesis on a fragment of a training simulator -- the environmental model. Since the Synthesis process supports the STARS vision of reuse, ARPA wanted to evaluate Synthesis with a pilot project. The pilot project leveraged assets from the Modular Simulator (Mod Sim, or USAF designation HAVE MODULE) project. The Mod Sim project aimed at creating a generic, tailorable simulator. The central Mod Sim concept is a partitioning of the training simulator into twelve separate segments which encapsulate functionality of the system and relate through defined, controlled interface messages.

STARS determined that the systems engineering knowledge captured in the Mod Sim assets, and the flexible architecture that derived from Mod Sim, constituted a sufficient basis for megaprogramming in the AVTS domain.

The domain for the pilot project could have been as large as Boeing S&TS's entire product line, or as small as a single instrument on the control panel of a simulator. It was determined that one of the twelve Mod Sim segments was the proper size for the study in that it would be large enough to provide scope for significant testing of Synthesis, while being small enough to be doable in the time available.

The Environment domain was selected because it best met these criteria. Environment includes the tactical and natural environments external to the simulated vehicle. Since the original Mod Sim implementation did not include the Environment segment there could be assurance that the segment's design and implementation were not preconceived.

The main conclusion was that the Synthesis process is effective when applied to real-time vehicle simulators. This process supports the main tenets of the STARS reuse philosophy. The Mod Sim architecture is suited for adaptability and reusability and when used in conjunction with the Synthesis process most of the work in defining the domain is already completed. Having the interfaces between segments defined greatly reduces the time required to assemble the assumptions and therefore is the key to a successful domain definition the decision model. This pilot effort concluded that Synthesis could be scaled to encompass the entire AVTS Domain.

Full Domain Analysis Pilot

When signing of the Memorandum of Agreement (MOA) between Advanced Research Projects Agency (ARPA) and Naval Air Systems Command (NAVAIR) for the STARS demonstration project was delayed, the STARS program manager decided to front load the Domain Engineering effort using Boeing S&TS. This became a pilot program to conduct initial domain analysis across the entire AVTS domain.

The pilot program adopted the Mod Sim partitions for the AVTS domain of twelve sub-domains: Propulsion

(PRO), Flight Controls (FC), Flight Station (FS), Instructor/Operator Station (IOS), Physical Cues (PHC), Environment (ENV), Visual (VIS), Navigation and Communication (NAV), Flight Dynamics (FD), Radar (RDR), Weapons (WPN), and Electronic Warfare (EW). Each sub-domain represented an area of expertise. Eight of these sub-domains apply to the T-34C FIT: Propulsion, Flight Controls, Flight Station, Instructor/Operator Station, Physical Cues, Environment, Navigation/Communication, and Flight Dynamics.

When the work required to define and specify the AVTS Domain was defined, a thirteenth sub-domain was identified: the AVTS sub-domain (in reality a super-domain of the other twelve) which contains information about the performance and fidelity of the simulator and the inter-process communication and sub-domain executives required by the software architecture.

Synthesis assumes as an operational concept a single commercial organization owning a product line. The organization defines its business goals and objectives as part of domain management. NAWCTSD, in co-operation with NAVAIR, has served as project owner, assisted by a team of contractors. We elected to proceed with the Domain Analysis by assuming the T-34C FIT was the first product in the Domain Evolution Plan.

At the conclusion of this phase, we had developed a first iteration (refined as needed) of the entire Domain Definition for the eight sub-domains relevant to the T-34C, plus the AVTS sub-domain. The Domain Specification, Process Requirements, and Product Requirements were also completed. Product Design was in draft form. These work products exceed the stated goal of supplying Software Requirements Specification level requirements, but fall short of the complete design desired.

Navigation/Communication Sub-Domain Pilot

After the partial trial DE effort, the project committed to a final pilot effort. The purpose of this effort was to gather further lessons learned and to provide technology transfer to the DE team intended for the full scale demonstration project. The focus of this effort was on a slice of the NAV sub-domain, particularly the radio aids components of Tactical Air Navigation (TACAN) and VOR.

A central difficulty with megaprogramming is determining how viable a domain may be. Obviously, the megaprogramming approach may not provide an economic payback for all product lines. An example of the difficulty of evaluating domain viability arose in the sub-domain. Casual consideration of the viability of the Navigation domain makes one somewhat skeptical due to its large number of inherent variabilities. For instance, the mode and power logic for aircraft equipment simulations will vary widely. These variabilities produced an extremely intricate network of decisions required to describe the operational characteristics of aircraft equipment, for example TACANs and VORs. However, moving past these surface difficulties one finds a rich set of commonalities in the underlying computations required for TACAN and VOR. For instance, geographic equations are the same whether they are used for TACAN or VOR navigation problem solutions. One hint of the hidden commonalities was the similarity of the data flowing in and out of these models. The domain engineering work products produced thus far in the project will continue to evolve, through additional iterations of Synthesis, into a more robust system that better leverages the lower level commonalities.

Pilot Lessons Learned

Analysis of variability and commonality results in generic architectural components that would not be created had a single point approach been taken

Use of metalanguage within components to implement adaptability breaks down the barriers that a generic Ada component provides. Metalanguage-based adaptability provides almost unlimited flexibility.

Automation of decision models to adapt code can be complex and costly. Careful metrics must/will be collected to provide feasibility information.

Commonalities and variabilities associated with this experience only involved the T-34, T-44, and T-45 series aircraft. As other aircraft are considered, the domain complexity will significantly increase during each iteration through the Synthesis process. Defining the domain (especially commonalities and variabilities) is time consuming, but with the right documentation and personnel experience, the process leads to many unexpected and helpful results.

The experience using the TACAN and VOR brought to light commonalities throughout AVTS. These include the computation of bearing, range, course deviation, and so forth. The variabilities are apparent when you consider the physical components and models. Switches, lights, and functionality differ between aircraft.

It is important that the domain decision model capture and group the choices between variations of domain instances according to some logical scheme. For example, instead of one large decision model called TACAN decision group, we decided to generate smaller decision groups, which consisted of logical groupings (e.g., Power, Self test, TACAN outputs, etc.).

We organized the decisions groups in a hierarchical structure to capture dependencies. This enabled the decision model to directly model the real world -- if the TACAN does not exist in this instance, then there is no need to exercise any of its related decision groups.

Synthesis requires identification of software components and their hierarchial relationship (product architecture) before beginning component design. This emphasis on careful partitioning of the software system is similar to that found in object-oriented design.

Synthesis tends to be life-cycle oriented rather than technique oriented. Therefore, traditional tools such as structural analysis, object-oriented analysis, program design language (PDL), pseudo-code, etc. were useful techniques for accomplishing Synthesis' goals.

TECHNICAL CHALLENGES

The project is now in a full scale demonstration phase. The objective of this phase is to complete sufficient domain engineering in a selected set of sub-domains, to an extent sufficient to support application engineering of a range of training simulators for T-series aircraft. The experiences derived from the pilot project have raised a series of concerns, or technical challenges that we anticipate will arise in the course of the full scale demonstration project.

Integrating Many Sub-Domains

One of the keys to our successful application of the DE process so far, has been our partitioning of the domain into related sub-domains. However, the pilot projects thus far have focused on only a small section of the domain or a partial DE cycle. Either focus faces fewer coordination problems than attempting full scale DE in many sub-domains. As we proceed, there is no reason to expect the DE process to proceed at the same rate in all of the sub-domains. One can envision DE progressing at different rate creating sub-domains that can not operate together. This problem has been called *domain integration.* Our response to this challenge has technical and management aspects.

Technical Aspects

Our technical approach within the domain contributes substantially to resolving this problem. First, the subdomains are *well-partitioned.* The pilot project based analysis (as well as the leveraged Mod Sim systems engineering legacy) suggests that only a very small fraction of subsystems may be reasonably assigned to multiple sub-domains. This fraction may be as low as 2 percent. Second, our technical approach depends on well-defined, controlled *interfaces.* The interfaces are defined in compilable Ada types. Each interface message is generated by a single subsystem (and as noted, a subsystem is associated with only one subdomain). Changes to these interfaces require agreement by the consuming sub-domains.

The third element of our technical approach that addresses the domain integration problem is that the sub-domains are highly *modular* and *localized.* The internal design of each sub-domain is required to conform to the Domain Architecture for Reusable Training Systems (DARTS), which specifies a hierarchial structure of modules (subsystems, components, and units) within segments. Figure 4 illustrates DARTS. Experience has shown that similar and related models are closely located. For example, the TACAN and VOR models are each encapsulated by DARTS components, and together (along with similar navigation equipment) in the Radio Aids subsystem. Strong localization means that functional changes tend to effect a small area of the system, rather than a broad effect.

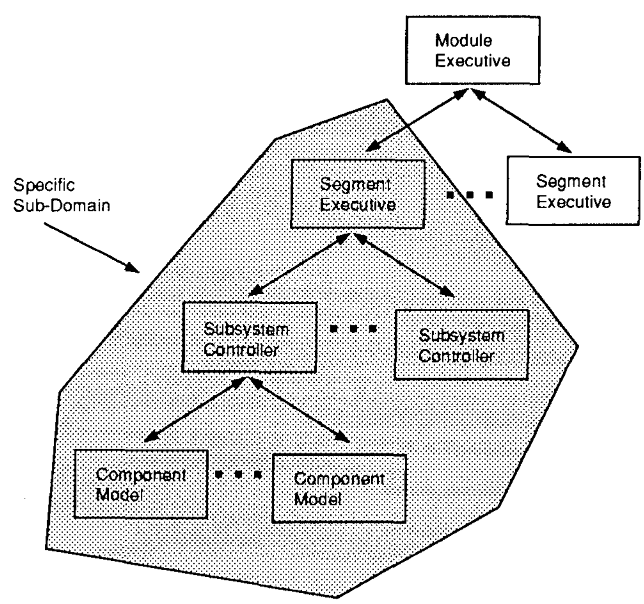


Figure 4: DARTS/Sub-Domain Relationship

The final element of our technical approach that addresses the domain integration problem is our adoption of *simulation strategies,* which guide the design and implementation of simulation models. These strategies capture the design decisions, which permeate throughout the simulator. The strategies create a common vision in every domain engineer's mind on how ubiquitous simulation problems will be addressed. Furthermore, they tend to reduce the number of nonessential variations to which the domain would otherwise have to adapt. The simulation strategies identified so far include: notification of electrical power, notification of malfunctions, assessing ownship damage, tracking real versus computed values, and tracking commanded versus current values.

Management Aspects

How can our project management approach help address the problem of domain integration? We expect there to be little sharing of subject matter experts between the different sub-domains, so cooperation between the sub-domains will be primarily a function of schedule and organization.

Our planning approach anticipates two distinct planning cycles: increments and iterations. *Increments* represent a management decision to invest sufficient resources to incorporate additional capability into the domain baseline. *Iterations* are complete or partial cycles through the Synthesis process. There may be many iterations within an increment. We do not plan to require sub-domains to proceed in concert for iterations -- but we do plan to require sub-domains to proceed in lock step through the increments. This is less structure than might first appear because the planned increments are fairly large and broad -- the current plan is one increment between now and October 1995. Figure 5 illustrates the increment/iteration relationship.

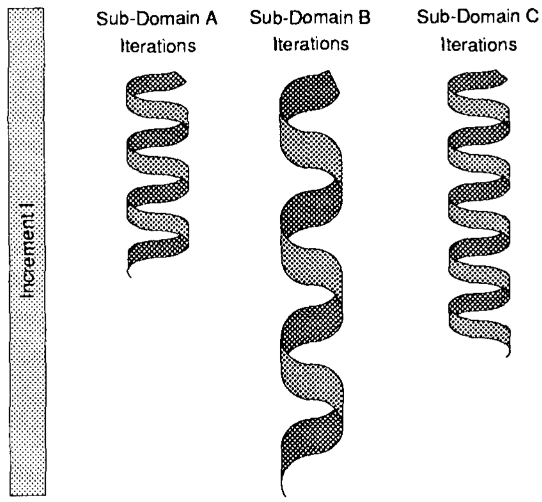


Figure 5: Increment/Iteration Relationship

In contrast to schedule, we expect our organization to provide substantial assistance. Figure 6 illustrates the current DE organization. This figure indicates that the very heart of the DE organization is the AVTS Domain Integration Team. This team exists to control changes that effect multiple sub-domains, and resolve such issues for the benefit of the entire domain. As such, its members have to maintain a systems mindset -- never seeking to optimize one component at the expense of the system. Notice also, that the twelve distinct sub-domains are grouped into systems groups, again following the principle of localization. The systems group leaders are required to have experience across the sub-domains within their jurisdiction.

Relating to Non-Domain Engineered Models

The demonstration project is focusing on *leveraged* reuse, i.e., products and product fragments uniquely built to be reused via defined processes. However, it is not clear how well this approach will support the

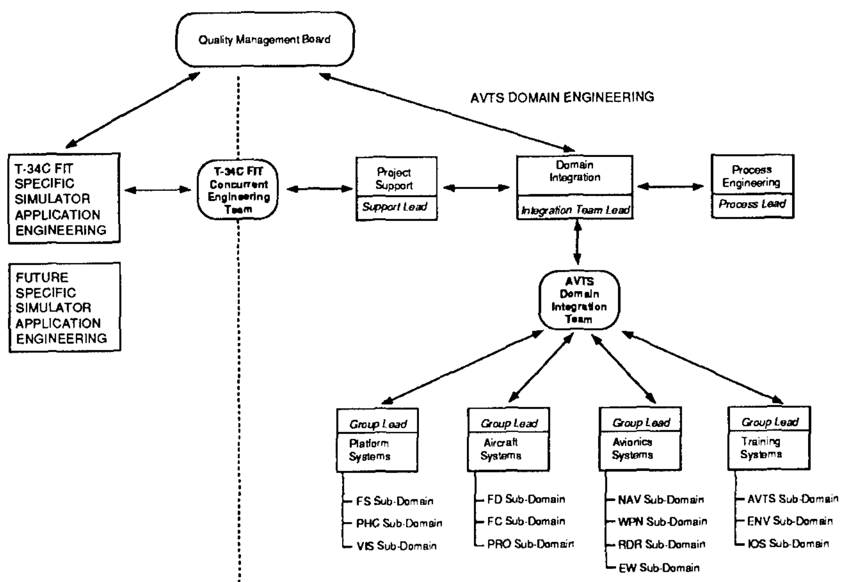


Figure 6: DE Organization

attempt to utilize *opportunistic* reuse, i.e., reuse based on the availability of existing assets. We should note that *re-engineering* of exists models does not in itself satisfy the AVTS requirement of providing *adaptable* models.

The central empowering concept in the domain is the insistence on a strong architecture. Only through such an architecture is leveraged reuse even possible, however, it does create specific limitations for opportunistic reuse; namely no component is usable which does not conform to the architectural constraints. If a component existed that conformed to the architecture, but was not developed through the DE process, it would be ideally suited for opportunistic reuse -- this however is very unlikely. Systems that impose a strong architecture for themselves are rare enough -- what is the chance that another system uses the derived architecture for AVTS?

Therefore, any component which is a candidate for opportunistic reuse will require some degree of re-engineering. In most cases, this may be relatively small. If an existing model cleanly encapsulates some functionality (say a single DARTS component), then the required re-engineering may be a simple as identifying and reconciling the interfaces. Once again, this is unlikely. There are as many ways to partition a simulator system as there are such systems, so even if a given model well encapsulates a problem within the context of its originating simulator, it probably crosses several partitioning lines in AVTS. The most likely case is that a single model will require partitioning to meet the AVTS partitions. In the worse case, a model may be based on assumptions that are dependent on data that in not available to an AVTS model. Such models would have to be re-designed.

None of these outcomes is satisfactory. Actual experience has shown that it is much better to use the existing assets as examples of what the leveraged sub-domains will have to provide rather than wholesale substitution. There is no substitute for actually performing the DE process, however, access to a rich collection of legacy assets can dramatically reduce the resources required and the risks incurred.

The most useful non-engineered models will be those that apply a strict separation between physics and controls -- a long standing tenet of modeling technique. Obviously the physical model is based on a very slow changing baseline (the earth) while the control model varies with each specific aircraft. Therefore, there is more opportunity for standardization and commonality with physical models. Models that cleanly divide between physics and controls tend also to have a clear interface, which again reduces the challenge of integrating them into the AVTS architecture.

Relating To A Real Product Acquisition Project

Megaprogramming makes no pretense of being a viable economic model unless *both* life-cycles are healthy. Most of this discussion has related to the challenge of domain engineering, but this is only the first part of the story. The products and processes of the domain have to be used by some actual application for there ever to be a payback from domain engineering. In fact, the envisioned situation would be several application project competing for instances from the domain. As discussed, the AVTS demonstration project has the T-34C FIT for its application project. While this simulator is very attractive from scalability and resource conservation aspects, it provides limited benefit in validating the domain engineering objectives for increment one: supporting a range of T-series aircraft with this first increment of domain engineering. Furthermore, the project lacks sufficient resources to achieve full scale domain engineering in all of the thirteen sub-domains in increment one.

The transition of megaprogramming of applications to the esoteric world of academics to the real world is a challenge that requires an incremental approach to the depth of domain engineering. The concept of full exploitation of all of the separate sub-domains at one time flies in the face of both synthesis and common sense. The Navy wishes to grow a product line of synthesis based simulators which would include the T-45, T-44, and the T-34C. The necessity is to rigorously model the domains in a manner that allows them to be expanded to include all of the vehicles in the product line. The process of concurrently synthesizing thirteen highly interactive sub-domains in a manner broad enough to cover three air vehicles would be extremely risky. The objective of first iteration is to approach the different sub-domains with varying level of domain development. The limit on utilization of the T-34C FIT as a prototype effort affords the DE & AE team a simplistic first iteration approach to synthesis.

The domain engineering team has decided to develop certain areas in a point solution method. For the point solution methods the sub-domain is concentrating on creating a model which is T-34C specific. Concurrent to the development of the point specific solution the sub-domain engineers are developing standard interfaces which will afford later growth. An example of this concurrent development is the sub-domain of propulsion. The T-34C engine is being specifically modeled and will have no variation in that model. However the variables, and structure of the model will allow the development of the domain in later increments in a manner which includes other gas turbine engines.

The domain engineering team is fully developing other sub-domains. These fully developed sub-domains will give the application engineer the ability to create new instances of the family if they has the correct data. One example of a fully developed sub-domain is flight dynamics. The full development of the domain will allow the application engineer to develop the correct flight dynamics model if they have the correct aerodynamics data. The growth of the sub-domain to include the other instances of the family line will require only the inclusion of the aerodynamics. The full development of other sub-domains is also requirement and data driven.

There are certain instances of the sub-domains which are not required for the product line of trainers. An example of these components are weapons. The T-34C, and T-44 do not employ weapons and for that reason the sub-domain of weapons is deferred to a later date.

The delineation of the methodology to synthesize domains was a critical step to gaining the capability of developing a synthesis based training system within the restraints of money and time. To assist us in the conversion of esoteric academics to robust reality we have physically separated the applications engineers from the domain engineers. While at first glance this may appear to separate the customer from the producer it appeared to us to be essential. It was apparent to us that if we had the AE & DE in close proximity to each other it would be easy for the AE team to drive the DE team to the T-34C solution and not to the product line solution.

Finally, we are committed to building a graphical representation of alternative cockpits in the product line. This *virtual cockpit* will give us the capability to exercise instances from the AVTS domain intended for platforms planned in the domain evolution plan, without incurring the cost and risk associated with use of actual simulator hardware. Furthermore, a virtual cockpit, graphically based, with a rapid prototype capability provides us a way to perform meaningful module tests and accomplish preliminary hardware/software integration.

Controlling Adaptation

Performing domain analysis is like riding a roller coaster -- there are highs when the domain seems perfectly suited to the process, and lows when it seems that the domain will never come together. It seems that the highs and lows are related to the depth of analysis in layers. Consider the following example.

At first glance, AVTS seems much too complicated to be a viable domain -- too many different kinds of aircraft and types of simulators. However, a little analysis reveals some important commonalities -- a simulator has some kind of flight dynamics, flight controls, and environment models, and instructor capabilities. However, further analysis (particularly in the avionics sub-domains) tends to re-agitate worries about the viability of the domain -- there are so many different equipment suites, much less the dozens of each kind of equipment (how many models of TACAN are there in military aircraft?). But once again, further analysis reveals the viability question -- yes, there are lots of different TACANs, but everyone of them requires essentially the same bearing calculation.

Another aspect of the problem is the reasonable decisions for an application engineer to make. For example, one can imagine an application engineer that can select the correct TACAN part number, but can they be expected know what the correct sweep rate is for a compass bearing pointer driven by a particular model TACAN is while searching for a station lock? While a TACAN designer may know such things, will they have the breadth of knowledge to make other decisions in the sub-domains, much less the AVTS domain at large. A decision model this complex would likely require the application engineering organization to import design level experts -- in which case, how does the domain show a return on investment?

The foregoing discussion illustrates that the specific level of detail captured in a decision model will play a large role in our sense of how viable a domain is -- and thereby what its likely return on investment will be. We have no satisfactory answers to this issue and plan to experiment with different solutions. Identification of a guiding principle for decision model complexity is a critical need for extending this approach to domains outside of the demonstration project.

Leveraging Extra-Domain Assets

A training simulator is (potentially) a very complex system. Upon casual reflection, one envisions the domain of training systems as a collection of various mathematical models interfacing with some kind of replicated crew interface controllable by some kind of operator/instructor device. While this is a reasonable simplified picture of the simulator, it does not begin to address the complete component set of the entire *system*. Figure 7 illustrates the broader view of a training simulator in which our domain exits.

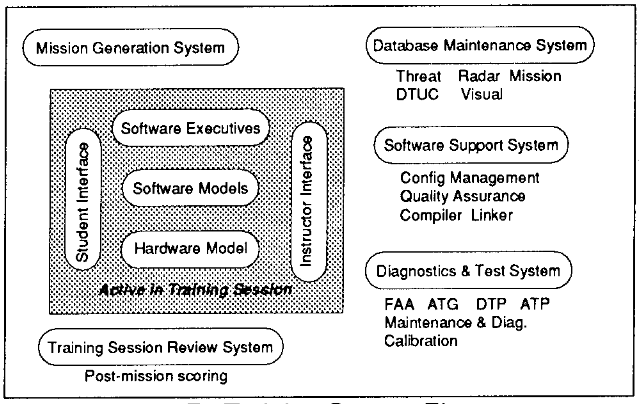


Figure 7: Training System Elements

While all of these system elements hold out good prospects for a reusable approach, the funding and scheduling constraints of the demonstration project would not permit examination of all possible elements. Therefore, we carefully considered what elements would constitute a minimal, yet complete system.

The operating hypothesis of the demonstration project is that while other system components such as mission generation, mission debriefs, etc., are significant resource consumers in training systems (and probably viable sub-domains), the appropriate focus for our work effort in the demonstration project are training system software elements active in a training session. There is no particular reason that those elements outside of the current AVTS domain boundaries could not be eventually included in the AVTS domain, or perhaps supported by a complementary domain.

Therefore, the demonstration project will have leveraged assets not created in the domain. For software assets, this does not present a serious problem. The SEE is fully capably of providing files that were not created in the course of domain engineering. The major problems will arise with acquisition of these assets, and interfacing to them. We expect that strong participation by NAWCTSD will open access to existing suitable assets. We plan to treat the interfaces to such assets as a routine variability of AVTS. Finally, the demonstration project has arranged for short run relaxation of some of the typical training system support requirements. For example, the increment one objectives include a sharply limited IOS requirement.

On the other hand, the non-domain hardware may present a substantially greater obstacle. Consider the simple example of the target computational system. The SEE exists on DEC engineering workstations using the VMS operating station. While some real-time applications have successfully utilized VMS (notably Manned Flight Simulator), it is unlikely to be a satisfactory answer for most applications in the domain. Applications will expect the SEE to provide cross-targeting capabilities, which are non-trivial to integrate.

A separate hardware problem is the stimulate/simulate debate. We plan to treat the decision to stimulate as a special case within the scope of the AVTS domain's variabilities. Of course, the various simulation approaches are routine aspects of the AVTS domain's variability.

CONCLUSION

The STARS goal is to increase software productivity, reliability, and quality via a paradigm shift; synergistically integrating computer-based support environments for modern software development processes and advanced software reuse concepts. We believe that the STARS approach may provide breakthroughs in the difficult problem of rapidly creating quality real-time simulator software. The STARS Program is entering the demonstration phase, with a Navy cosponsored effort to apply the STARS approach to the construction of a T-34C Flight Instrument Trainer.

However, these capabilities are not easy to deliver. Some of the difficulties in creating a domain of simulator software are: (a) the size and scope of the problem area (b) non-specific variabilities between users, and (c) the not-invented-here syndrome. The real world considerations of funding and schedule constraints, combined with constantly evolving requirements apply to the demonstration project, just as they do to more traditional projects. Never-the-less, experience deriving from the series of megaprogramming pilots has created high expectations for a technical success.

REFERENCES

[Ausnit85] Ausnit, Charles, et al. *Ada Reusability Guidelines.* April 1985. SoftTech Incorporated. Waltham, MA.

[Boehm92] Boehm, B. and B. Scherlis. "Megaprogramming", *Proceedings of the DARPA Software Technology Conference.* 1992. Meridien Corporation. Arlington, VA.

[NATOPST-34C88] Department of the Navy. 1988. *NATOPS Flight Manual Navy Model T-34C Aircraft.* NAVAIR 01-T34AAC-1. Naval Air Systems Command. Washington, D.C.

[VCOE93] Software Productivity Consortium Services Corporation. 1993. *Reuse-Driven Software Process Guidebook.* SPC-92019-CMC. Virginia Center of Excellence. Herndon, VA.

The Mapping of Object-Oriented Design to Ada Implementation

John Glaize, Staff Scientist
CAE-Link Corporation
Binghamton, NY

ABSTRACT

Object-oriented development (OOD) methodology is rapidly emerging as the technology of choice for producing maintainable and reusable software. The Ada programming language was formulated to accomplish these same objectives. Therefore it becomes critical to develop proper mappings for implementing the structures of OOD into the Ada language. A "proper" mapping is one that exploits the inherent advantages of both OOD and Ada, preserves a natural and understandable correspondence between the design and the implementation phases of a project, and minimizes redesign and rework costs by easing the transition from design to implementation. As companies seek to adopt OOD and Ada, there are usually significant costs associated with training the engineering staff to use these methodologies. These costs and their attendant risks can be greatly ameliorated by choosing mappings that simplify and improve consistency between design and implementation. This paper investigates mappings of such OOD concepts as classes, objects, generalization-specialization structures, whole-part structures, and object services (methods) to Ada 83 programming constructs. Judicious application of these mappings can result in significant savings in training, development, maintenance, and reuse costs, and ease the transition to OOD and Ada.

About the Author

John Glaize is a Staff Scientist and Ada Specialist at CAE-Link Corporation where he has held various management and technical positions over the last 20 years. His primary task for the last several years has been to develop and refine a real-time Ada architecture and software development methodology to maximize the production of high-quality maintainable and reusable software. He holds a B.S. degree in Mathematics from Carnegie-Mellon University and an M.B.A. from Binghamton University.

The Mapping of Object-Oriented Design to Ada Implementation

John Glaize, Staff Scientist
CAE-Link Corporation
Binghamton, NY

INTRODUCTION

In this post cold-war era, defense contractors have become, by necessity, increasingly aware of the economics of software development. It is no longer sufficient for software to provide the correct functionality, but it must also be cost-effective to produce, maintain, and modify. To maintain this cost-effectivity across a variety of projects, the software should also be reusable. To support these goals, many software engineering technologies have been developed by the software community, and embraced by the defense industry. Indeed, much effort and cost has been expended to ensure that these technologies are successful. Of particular concern to many companies, however, is that different technologies have been developed for different phases of a project, and there often does not exist a clear and effective correlation and transition from one technology to the next.

Two such technologies are Object-Oriented Development (OOD) and the Ada programming language. Both OOD and Ada were developed to promote such software engineering concepts as encapsulation, abstraction, information hiding, and understandability. Both have achieved these goals to a significant extent. However, OOD is used during the design phase of a project and Ada is used during the implementation phase; often a "disconnect" between the two phases and the two techniques can result. CAE-Link Corporation, as part of the ongoing development of the very large-scale B-2 Aircrew Training Device (ATD), has explored mapping methodologies to help make the transition from design to implementation more natural, understandable, and cost-effective.

OOD AND ADA

OOD is a software development methodology based on describing and partitioning a software system into a collection of Classes of Objects. An Object Class is an abstraction of the real world that focuses on the germane aspects of the problem domain and the software system's responsibility for solving the problem. An abstraction consists of both the attributes of a Class of Objects (i.e., the data that describes the state of the Objects), along with all services that process or have knowledge of that data. Due to this closer association of problem space and solution space, an object-oriented design is generally more understandable, and the effects of changes more isolated. The OOD methodology realizes these advantages especially well in data-driven applications, where modifications are generally manifested in changed data specifications. An ATD is largely a data-driven application, and thus is well suited to this methodology.

In addition to basic Classes, OOD involves describing the solution space in terms of "structures" of Classes. A Whole/Part structure is analogous to an "assembly" structure, where a larger Class (the Whole) contains a set of smaller Classes (the Parts). A Generalization/Specialization (or Gen/Spec) structure defines a "general" Class (such as a Vehicle), and a set of specializations of this general Class (such as an Automobile or a Truck). Each specialization can take on the attributes and services of the generalization (this is referred to as "inheritance") or extend the generalization by adding to or modifying the attributes and services.

The Ada programming language bears a fairly natural relationship to OOD. Ada 83 has been classified as an "object-based" language because it promotes the package construct that greatly supports abstraction and information hiding. In fact, Booch maintains that "Abstraction and information hiding form the foundation of all object-oriented development"[1]. However, even though OOD and Ada may appear to be compatible partners in software development, it is extremely important that a mapping be formulated to effect a smooth transition between design and implementation.

IMPORTANCE OF A MAPPING

First, one might question why design and implementation should even use different technologies. In particular, why couldn't Ada be used as a Program Design Language (PDL) during the design phase as well as the implementation language?

Link's experience in this area indicates that design and implementation are really distinct activities with different goals and different products. During the design phase, a careful correlation of the solution space to the problem space must be maintained to ensure that all requirements are being met. A graphical design representation, such as Class and Object diagrams, allows the design to be "seen". It is easier to determine that the design is complete, consistent, and solves the problem.

When the same language is used as a PDL and implementation language, Link's experience is that the design is done at too low a level, becomes indistinguishable from the code, and requires too much maintenance (i.e., code changes tend to require updates to the design representation, PDL). The result is that the design and implementation are no longer distinct activities, but become merely different representations of the same information. To avoid these pitfalls and to ensure that the design phase and the implementation phase both retain value, different technologies for the two phases are indicated.

However, this means that it can be difficult to correlate the design structures to the implementation structures, and therefore difficult to assess whether or not the implementation faithfully represents the design. In addition, the implementation engineers may be uncertain as to the usage of various Ada constructs (for example, how should programs and data be named, what should Ada types be used to represent, and what should be included in Ada packages?). Link's experience is that without precise direction, many different and inconsistent coding styles can arise. This makes it more difficult to understand the entire system, and impedes the interfacing and the reuse of the various software components. The need to maintain the distinction between design and implementation, but to also closely correlate these two activities gives rise to the importance of a mapping.

In addition, as indicated above, Ada 83 is classified as an "object-based" language rather than a truly "object-oriented" language. This means that Ada does not embody all of the features required to implement all OOD constructs. Most notably, Ada 83 does not have the "inheritance" capability which is so much a part of the Gen/Spec structure described earlier. For reasons such as this, the transition from some OOD constructs to Ada implementation is not obvious.

Finally, unless the design methodology dovetails nicely with the implementation methodology, it may be necessary to formulate and conduct additional training courses to instruct the engineers in the transition. This increases the cost of software development, and is counterproductive to the cost-effectivity of the development process.

Note that even if CASE tools are used to automatically generate Ada code based on the OOD design representations, a mapping should still be formulated so as to evaluate and/or tailor the code generation capability of the CASE tool.

MAPPING METHODOLOGY

OOD techniques allow the engineer to describe the design solution with both static and dynamic concepts. The static concepts are Class and Object structures that identify the "players" in the solution space in terms of what they are, i.e., their attributes and the services they perform. The dynamics indicate how the players interact to solve a particular problem in terms of the interfaces between them and the manner in which their services are invoked.

Due to size considerations, this paper confines its focus only to the static concepts of OOD. Table 1 illustrates an overview of the primary OOD static structures, and a methodology that has been investigated by CAE-Link for mapping these structures into an Ada implementation.

CAE-Link uses Class and Object diagrams[2] to represent the design of the static structures. Each diagram depicts the name, the attributes, and the services for each Class. This paper explains the mapping by outlining a "construction methodology" that provides a step-by-step process for taking each type of OOD structure into its corresponding Ada implementation. Then, to further illustrate how the process works, a sample OOD diagram is shown, along with the resultant Ada code. Rationale is also given to explain why the mapping was done this way.

The examples given herein are for illustration purposes only, and do not necessarily represent a working, real-world design. For the sake of brevity, some attributes and services have been abbreviated or eliminated. In some of the Ada examples, complete and compilable

Table 1
OOD Static Structures to Ada Implementation Overview

OOD Structure	Ada Implementation
Class	An Ada package containing the necessary data type(s) to define the attributes of the Class, along with subprograms (procedures and functions) that define the services of the Class.
Whole/Part Class Structure	A fundamental question for this structure is whether both the Whole Class and the Part Classes are implemented, or just the Part Classes. This question is addressed fully in the Whole/Part section of the paper, but both Whole Classes and Part Classes, when implemented, are implemented as Ada packages as described under "Class" above.
Gen/Spec Class Structure	Both Gen Classes and Spec Classes are implemented as Ada packages, as described under "Class" above.
Object	A declared Ada data object. The type of the data object matches the type definition(s) contained in the Ada package that implements the Class of which this Object is a member.

code does not appear, only enough to illustrate the concepts being presented.

CLASS

The construction methodology for an OOD Class is as follows:

- o The Class is implemented as an Ada package whose name is the same as the Class name in the diagram (e.g., engine_inlet in Figure 1). The use of a package causes the Class attributes and services to be encapsulated together within the same coding structure in order to promote maintainability and understandability. Using the same name for the Class and the Ada package ensures that the naming of Classes is consistent across all project software, and that the correspondence between design Class diagrams and the Ada packages which implement them is self-evident.

- o The package specification contains a record type that represents the data attributes of the Class. The type is named "data" and contains, as record components, all attributes of the Class. The type is implemented as a <u>private</u> type to support information hiding and to utilize the checking features of the Ada compiler to ensure that only the services of the Class have knowledge of the data structure. The reason for naming the type simply "data" instead of something apparently more descriptive like "inlet" is that the type is always used in the context of the package name (i.e., "engine_inlet.data"), and this avoids the repetitive naming structure such as "engine_inlet.inlet". Making the data type a record ensures that all attributes of the Class are encapsulated.

- o The package may contain auxiliary types used to construct the record named "data". An example of such a type is the type named "compartments" that is shown in Figure 4.

- o The services of the Class are implemented as Ada subprograms (procedures and functions). The name of each subprogram is the same as the name of the service in the Class diagram that it implements so as to preserve correspondence between design and implementation. The specification for each service is in the Ada package spec and the body is in the Ada package body. Typically, an Object of the Class is passed to a service as a parameter for servicing. For example, an Object of the Class engine_inlet, such as left_outboard_engine_inlet illustrated in Figure 10, is passed as a parameter to the service update_inlet_characteristics for servicing. This helps maintain the correspondence between a

Class, Objects of the Class, and the Class services that operate on the Objects.

Figure 1 illustrates a Class named engine_inlet which has four attributes: fan_pressure, fan_temperature, fan_airflow, and pressure_ratio_ambient, and one service: update_inlet_characteristics.

engine_inlet
fan_pressure fan_temperature fan_airflow pressure_ratio_ambient
update_inlet_characteristics

Figure 1
Sample OOD Class Diagram -- Engine Inlet Class

Figure 2 illustrates the Ada package specification that implements the engine_inlet Class. It shows that the name of the package matches that of the Class, that the attributes of the Class are encapsulated in a data record implemented as a private type, and that the services of the Class are implemented as Ada subprograms, complete with their calling sequences. Each time a service, such as update_inlet_characteristics, is invoked, an inlet is passed to the service as a parameter for servicing. The logic for the services is implemented in the corresponding package body (not shown).

WHOLE/PART STRUCTURE

As stated earlier, a Whole/Part structure resembles an "assembly" structure, where a larger Class (the Whole) contains a set of smaller Classes (the Parts).

The construction methodology for a Whole/Part Structure is as follows:

- o Both the Whole and the Parts are Classes, and therefore are implemented as Ada packages using the same methodology as for OOD Class.
- o The only time that a Whole Class on a Class Diagram is not actually implemented in Ada is when that Class has neither attributes nor services of its own (i.e., all attributes and services are embodied in the Part Classes). An "empty" Class like this (sometimes referred to as a "placeholder") should not appear in a

```
With standard_engineering_units; use standard_engineering_units;
With engine_environment;
Package engine_inlet is
   Type data is private;

   Procedure update_inlet_characteristics(
      for_the_inlet               : in out engine_inlet.data;
      using_fan_inlet_flow        : in     pounds_per_second;
      and_engine_environment      : in     engine_environment.data);

Private
   Type data is record
      Fan_pressure                : pounds_per_square_inch;
      Fan_temperature             : degrees_rankin;
      Fan_airflow                 : pounds_per_second;
      Pressure_ratio_ambient      : ratios;
   end record;
end engine_inlet;
```

Figure 2
Sample Ada Implementation -- Engine Inlet Class

good design, and therefore, the typical condition is that all Classes in a Whole/Part structure are implemented.

- o When the Whole Class of a Whole/Part Structure is implemented, the record named "data" in the Ada package may contain not only the attributes of the Whole, but also instances of the Part Classes. For example, the fuel_tank Class, whose implementation is depicted in Figure 5, contains Compartment and Baffle Parts in its data record.

  In this example, Compartment is a Class which has no parts, and therefore, its data record contains only its attributes, as depicted in Figure 4.

- o Services of a Whole Class may invoke services of the Part Classes as illustrated in Figure 6 (in this example, the fuel_tank service set_demanded_fuel_quantity invokes the compartment service set_demanded_fuel).

Figure 3 illustrates a Whole Class, fuel_tank, that has its own attributes and services, such as total_quantity and set_demanded_fuel_quantity, as well as two Part Classes, compartment and baffle. Since all Classes possess their own attributes and services, they are each implemented in Ada.

Figure 4 illustrates a sample implementation for one of the Part Classes, compartment. The implementation follows the construction methodology for a Class, and looks quite similar in structure to the engine_inlet Class in Figure 2 with its data record and subprograms. The implementation of the baffle Class (not shown) has an equivalent structure.

Figure 5 illustrates several features of the Whole/Part Ada implementation. The package fuel_tank contains some auxiliary types, such as compartments, that are used to construct the "data" record. The "data" record also contains instances of the Part Classes, such as compartment_set and baffle_set. In addition, Figure 5 illustrates how other features of the Ada language, such as the discriminant number_of_compartments, can be used to configure the "data" record. When instances of the Class fuel_tank are defined, individual fuel_tanks can be configured with different numbers of compartments.

Figure 6 illustrates the fact that services of the Whole Class, such as set_demanded_fuel_quantity, can be implemented to invoke services of the Part Classes, such as compartment.set_demanded_fuel. Note that due to the fact that the "data" record for the compartment Class is implemented as a private type, the fuel_tank service must use a compartment service if it wishes to service a compartment. This tends to isolate any compartment design changes to the compartment Class, and not propagate such changes to the fuel_tank Class.

GEN/SPEC STRUCTURE

Gen/Spec Structures are of particular usefulness when implemented in a language that supports inheritance and dynamic binding. Since Ada 83 does not support these features, the construction of Gen/Spec structures is somewhat awkward in Ada, and their usefulness is impaired. However, it is still possible to construct such classes.

The construction methodology for Gen/Spec Classes is:

- o Both the Gen and the Specs are Classes, and therefore are implemented as Ada packages using the same methodology as for OOD Class.

- o The record named "data" in each Ada package that implements a Spec Class contains a component which is an instance of the Gen Class. That component can then be serviced by the Gen Class's services.

- o This methodology should not be done for more than one level of Spec Structures below the Gen Structure since Ada withing only extends to one level.

Figure 7 illustrates a Gen Class, radio, and two Spec Classes, tacan and uhf_vhf. Each of the Spec Classes "inherit" the attributes and services of the general radio, while extending these general features with attributes and services of their own.

Figure 8 illustrates that the Gen Class is implemented following the construction methodology of a typical Class, with a "data" record representing the attributes, and subprograms implementing the services.

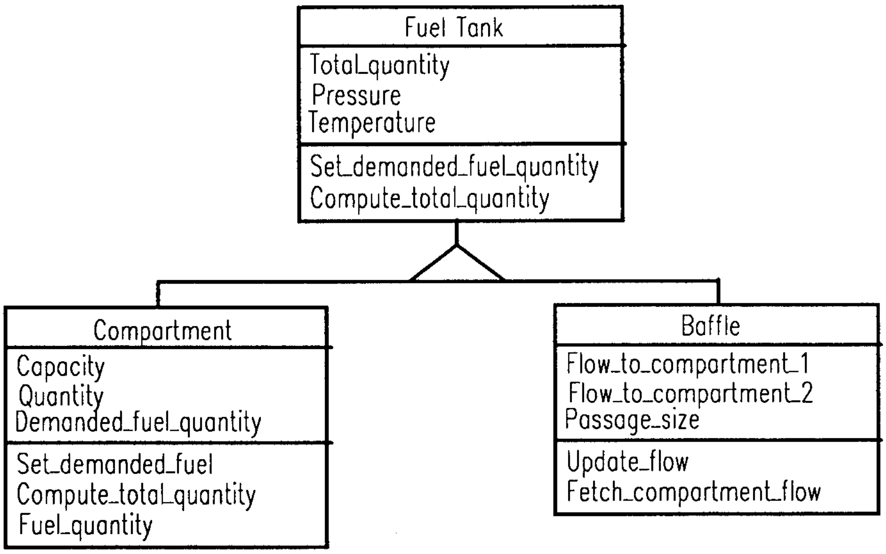


Figure 3
Sample Whole/Part Structure -- Fuel Tank

```
With standard_engineering_units; Use standard_engineering_units;
Package compartment is
   Type data is private;

   Procedure set_demanded_fuel
      (for_the_compartment          : in out compartment.data;
       using_the_demanded_fuel      : in gallons);

   Procedure compute_fuel_quantity
      (for_the_compartment          : in out compartment.data;
       using_the_aircraft_attitude  : in degrees;
       the_manifold_fuel_flow       : in gallons_per_minute);

   Function fuel_quantity
      (for_the_compartment          : compartment.data) return gallons;

Private
   Type data is record
      Capacity                      : gallons;
      Quantity                      : gallons;
      Demanded_fuel_quantity        : gallons;
   end record;

end compartment;
```

Figure 4
Sample Part Class Implementation -- Compartment

```
-----------------------------------------------------------------------
With compartment;
With baffle;
With standard_engineering_units; Use standard_engineering_units;
With equipment_status_types; Use equipment_status_types;

Package fuel_tank is
   Type compartments is array (positive range <>) of compartment.data;
   Type baffles is array (natural range <>) of baffle.data;

   Type data (number_of_components         : positive) is private;
   Procedure set_demanded_fuel_quantity
        (the_fuel_tank                     : in out fuel_tank.data;
         the_demanded_quantity             : in gallons);

   Procedure compute_tank_total_quantity
      (for_the_fuel_tank                   : in out fuel_tank.data);
Private
   Type data  (number_of_compartments      : positive) is record
      Total_quantity                       : gallons;
      Pressure                             : pounds_per_square_inch;
      Temperature                          : degrees;
      Compartment_set                      : compartments (1..number_of_compartments);
      Baffle_set                           : baffles (1..number_of_compartments-1);
   end record;

end fuel_tank;
-----------------------------------------------------------------------
```

Figure 5
Sample Whole Class Structure -- Fuel_tank Specification

```
-----------------------------------------------------------------------
Package body fuel_tank is
   Procedure set_demanded_fuel_quantity
             (the_fuel_tank                : in out fuel_tank.data;
              the_demanded_quantity        : in gallons) is
      Compartment_demanded_fuel            : gallons;
   begin
      For compartment_index in the_fuel_tank.compartment_set'range loop
         Compartment_demanded_fuel :=
            the_demanded_quantity / the_fuel_tank.number_of_compartments;
         Compartment.set_demanded_fuel
            (for_the_compartment =>
                 the_fuel_tank.compartment_set (compartment_index),
             using_the_demanded_fuel => compartment_demanded_fuel);
      end loop;
   end set_demanded_fuel;

   -- Include implementations of other services here.
end fuel_tank;
-----------------------------------------------------------------------
```

Figure 6
Sample Whole Class Structure -- Fuel_tank Body

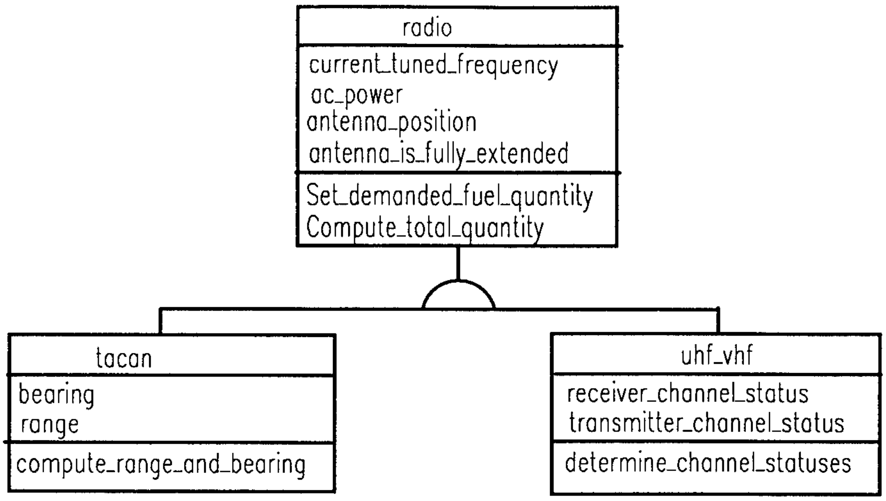


Figure 7
Sample Gen/Spec Structure -- Radio with TACAN and UHF-VHF

```
-----------------------------------------------------------------------
With std_eng_types; use std_eng_types;
Package radio is
   Type data is private;
   Procedure compute_current_frequency  ...

   Procedure determine_antenna_status ...
Private
   Type data is record
         current_tuned_frequency      : hectohertz;
         ac_power                     : watts;
         antenna_extended_position    : inches;
         antenna_is_fully_extended    : boolean;
   end record;

end radio;
-----------------------------------------------------------------------
```

Figure 8
Sample Gen Class Ada Implementation -- Radio Class

Figure 9 illustrates an implementation that results in an effect somewhat similar to inheritance. The "data" record contains a component, basic_radio_characteristics, that includes the attributes of the radio Class within the tacan Class. Likewise, the tacan has a procedure compute_range_and_bearing that implements its own service, while using the Ada renames feature to "include" the services of the radio in the tacan and make them available to users of the tacan Class. The implementation of uhf_vhf (not shown) is done similarly.

OBJECTS OF A CLASS

The construction methodology for an OOD Object is as follows:

- o An Object is an instance of a Class. It is implemented as a data declaration typically in a separate package from the package implementing the corresponding Class.

- o The declaration is typically a data object whose type is the record type "data" in the Class package, or an array of data objects that represent a collection of Objects.

- o The package that contains the declaration of the Objects is separate from the package(s) that implement the Classes because the Classes typically define a reusable abstraction that pertains to many applications, whereas the Object declarations define the number and identity of the instances that pertain to a particular application.

Figure 10 illustrates the Objects of the Class "inlet" of Figure 2. The Class is more general, reusable, and applicable to many aircraft. The Objects, as implemented, apply to a 747 aircraft, and indicate that there are four inlets and have particular designations (e.g., left_outboard_engine inlet). This is how the Object implementation describes the number and identity of the instances of a Class. Figure 10 also illustrates that the Objects can be implemented as individual data objects or as an array (collection) of data objects.

```
With radio;
Package tacan is
   type data is private;
   Procedure compute_range_and_bearing
        (for_the_station            : tacan.data;
         using_the_target_latitude : degrees;
         and_the_target_longitude  : degrees);

   Procedure compute_current_frequency
        (for_the_radio              : in out radio.data;
          using_the_tuner_position : in dial_positions) renames
                                     radio.compute_current_frequency;

   Procedure determine_antenna_status ...
                              renames radio.determine_antenna_status;
Private
   Type data is record
      basic_radio_characteristics : radio.data;
      bearing                     : degrees;
      range                       : nautical_miles;
   end record;
end tacan;
```

Figure 9
Sample Spec Class Implementation -- TACAN Radio

```
Package seven_forty_seven_engines is

   left_outboard_engine_inlet      : engine_inlet.data;
   left_inboard_engine_inlet       : engine_inlet.data;
   right_inboard_engine_inlet      : engine_inlet_data;
   right_outboard_engine_inlet     : engine_inlet_data;

-- Alternatively, the engine inlets could be implemented as an array.

   type engine_list is (left_outboard_engine, left_inboard_engine,
                        right_inboard_engine, right_outboard_engine);

   type engine_inlet_set is array (engine_list) of engine_inlet.data;

   inlets : engine_inlet_set;

end seven_forty_seven_engines;
```

Figure 10
Sample Object Implementation -- Engine Inlets

CONCLUSION

A combination of customer requirements and the pressures of the marketplace are forcing companies to become more and more cognizant of technologies that manage costs and produce higher quality software. In concert with this initiative, the software development process must include a precise, natural, and understandable transition between the design and implementation phases of a project.

Often the design methodology is OOD and the implementation language is Ada. Companies adopting OOD have been more easily able to produce a design that can be visualized, validated for consistency, and cross-checked against the requirements. By isolating the effects of changes and correlating to the real world, this design is also more maintainable and reusable. Use of the Ada language dovetails nicely with OOD to promote abstraction and information hiding, while keeping the design activity separate from the coding activity, and obviating the need for additional training. All of this adds up to higher quality, reusable software at a more reasonable cost. Put simply, this results in more company profit!

This paper has shown the necessity for a mapping between the design and implementation phases of a project, and has presented a representative mapping for translating OOD design to Ada, along with a discussion of some of the rationale that has led to the formulation of that mapping. CAE-Link has successfully used this mapping to produce better software. Companies must recognize the value of incorporating a design-to-implementation mapping into a documented, consistent, and cost-effective methodology for software development.

REFERENCES

[1] Booch, Grady Software Engineering With Ada, Second Edition, The Benjamin/Cummings Publishing Company, Inc. 1987

[2] Coad, Peter and Yourdon, Edward, Object-Oriented Analysis, Second Edition, Yourdon Press Computing Series, Prentice-Hall, 1991

INTERFACES AND THEIR MANAGEMENT IN A LARGE ADA PROJECT

Walter E. Zink, Senior Systems Engineer
and
Richard E. Poupore, Systems Engineer
CAE-Link Corporation, Binghamton, NY

ABSTRACT

The Department of Defense continues to require that Ada be the sole programming language for all new software related projects. In addition, these new projects are expected to achieve higher levels of maintainability from a software perspective. Ada and its related compilation/software engineering issues have given interfaces and their management a whole new perspective. In today's environment of dwindling defense dollars, extensive rework during the development or maintenance phase of a project due to interface changes, is prohibitive. Therefore, it is crucial to the success of large Ada projects to address interface issues from the highest perspective. For example, in a simulation environment, as the real-world device changes, the simulator must remain concurrent to provide maximum training benefit. These changes often result in changes to interfaces. In order to keep pace with the development and subsequent upgrades, it is necessary to provide reliable, maintainable and flexible interface structures. By combining a successful software architecture, a database-driven interface management tool and auto-generated connection software, major interface updates can be made in a timely and efficient manner. Experience has shown that with the proper interface design strategy, maximum cost savings can be realized over the entire life cycle of the simulator. An approach to interfaces, their management and connection software is discussed.

ABOUT THE AUTHORS

Walter E. Zink Sr. is currently a Senior System Engineer at CAE-Link Corporation, where he has held management and technical positions since 1986. His current effort is devoted primarily to system-level software design of the Real-Time Simulation Environment that supports the Ada architecture in use at Link. Prior to that, Mr. Zink was employed by the General Electric Company for twelve years, where he held various technical and management positions. His principal activities there were in the areas of Computer Based Instruction and Artificial Intelligence/Expert Systems. Mr. Zink holds a B. S. in physics from Harding University. Other publications includes a paper entitled "The Challenges of Developing A Real-Time Environment in Ada", presented at the 13th Interservice/Industry Training Systems Conference (1991).

Richard E. Poupore is a Systems Engineer at CAE-Link Corporation, where he has been employed since 1987. He is currently working on several system and sub-system level tasks involving inter-task interfaces, interface management and vendor systems interfaces. Mr. Poupore's involvement in these tasks has included system design, development and testing. He was primarily responsible for the design and development of the Shared Memory Management system. He has also been involved with a real-time debugger, the sequencer software and a real-time, no-wait I/O system. Mr. Poupore holds a B. S. degree in computer engineering from Rochester Institute of Technology.

INTERFACES AND THEIR MANAGEMENT IN A LARGE ADA PROJECT

Walter E. Zink, Senior Systems Engineer
and
Richard E. Poupore, Systems Engineer
CAE-Link Corporation, Binghamton, NY

INTRODUCTION

This paper provides an overview of the management of interfaces among software elements and its significance in the overall life cycle process. To do this, the paper begins with requirements imposed on the B2-Aircrew Training Device (ATD) and a view of the simulation computer environment. This is followed by the issues associated with interfaces and the project's requirements. An interface management system is discussed. This system includes a tool to manage interfaces and generate interface data movement software. The system also includes the design of the generated software. The paper concludes with a discussion of the benefits derived from this interface management system.

REQUIREMENTS

The B2-ATD is a high fidelity training device. The training device is required to provide not only a realistic copy of the cockpit but also a realistic training environment. To provide this, a substantial amount of Ada code is needed. The B2-ATD currently has almost two million lines of code.

The B2-ATD was one of the early Air Force Ada projects. The Air Force felt that traditional simulation architecture would not be sufficient. They required that a structural model, as defined by Carnegie Mellon University's Software Engineering Institute (SEI)[1], be used for the software architecture. Part of the required structural model is a software data bus. As interfaces move across the data bus they must remain uniform with respect to time and each other. A Computer Software Component (CSC) must be able to retrieve data from another CSC in a consistent manor. A CSC is defined as a simulated system such as engines.

As with most military projects, the Air Force placed the requirement that all CSCs be independently testable. An engineer should be able to write tests in a development environment and use those tests on the simulator. The engineer should also be able to insert the CSC into the simulation load and be able to test it when not all of the CSCs on which it depends are in the load.

The Air Force also required the ability to keep the software current with the B2 air vehicle. If a system is added or modified in the air vehicle, a corresponding change must be made to the simulator. These changes need to be made in a fast and accurate way. To support this, interfaces must be easy to update.

The B2-ATD is also required by DOD-STD 2167A[2] to provide documentation of interfaces. An Interface Design Document (IDD) that reflects interfaces of the simulator must be provided.

ENVIRONMENT

Hardware

The main computational engine consists of five systems, each with four processors. Each system has a base rate. The processors in that system run at the base rate or some even submultiple of that rate. The systems run at different base rates depending on the simulated systems running on them or the peripheral devices hanging from them. The computational engine is where most of the simulation software executes. It also contains most of the over thirteen thousand interfaces. A pair of workstations power the Instructor Operator System (IOS), the man-machine interface for the instructor. Several peripheral devices are connected to the computational engine to perform a variety of functions. These include an Image Generator (IG), a Digital Radar Land Mass System (DRLMS), interfaces to B2 On-Board Computers (OBCs) and a VME-based hardware interface system. The simulation computer complex is shown in Figure 1.

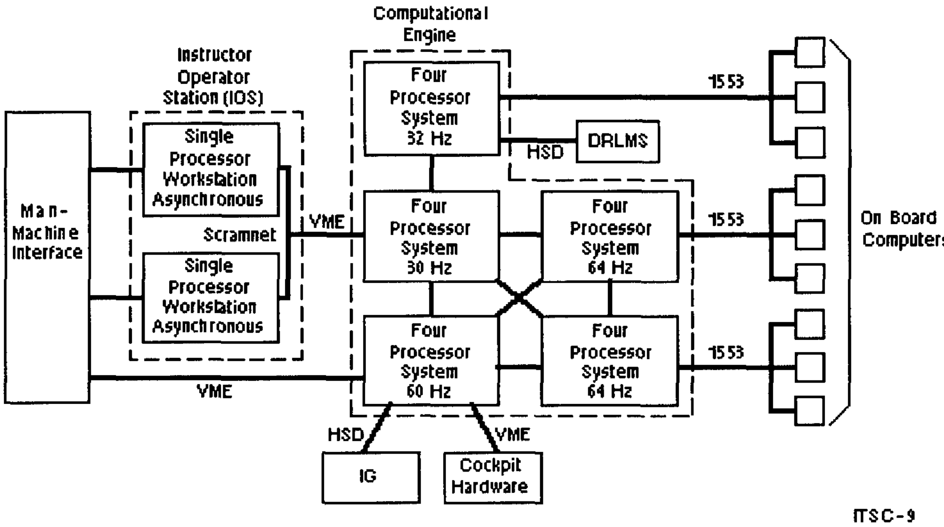


Figure 1 The Simulation Computer Complex

Software

The Ada architecture provides a unified and consistent basis for developing software which operates in a complex hardware environment. It is consistent with SEI's Air Vehicle Architecture[3]. It is comprised of several classes of "elements", each element performing a specified function in the overall makeup of the system and subsystems. It provides stringent data and control flow as well as the ability to test on immature or partially defined system.

The architecture is based around the autonomous CSC, its immediate environment and its relationship to other CSCs. Each CSC is comprised of object definition package(s), object declaration package(s) and a CSC control manager. Figure 2 depicts a typical CSC, its process and data flow. The figure uses a graphical notation described in "A Graphical Notation For Software"[4].

The definition package(s) provides the abstract for the CSC's objects and their control. It provides types that define the object's internal data structures. The definition package(s) also contains functions and procedures that provide the basic functionality for manipulation of the objects. The declaration package(s) provides instances of the CSC's objects. The control manager is responsible for determining how the CSC will respond to any given simulation state and the update of the CSC's objects. It invokes the operations in the definition package(s) and thereby controls the execution of the CSC.

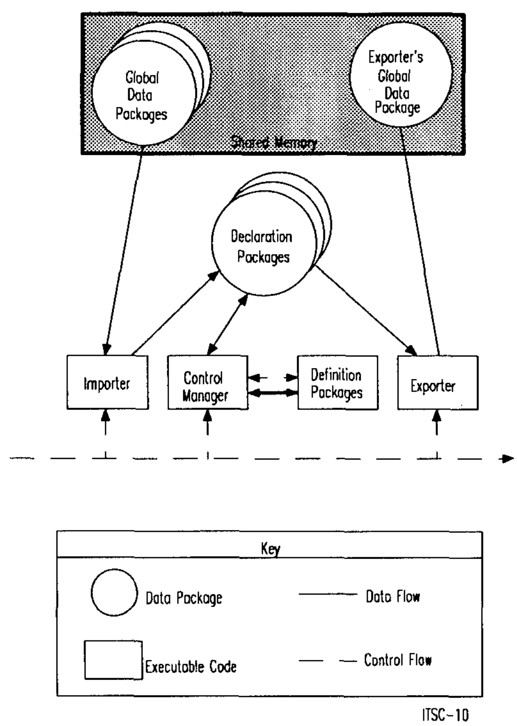


Figure 2 A Typical CSC

Wrapped around the CSC are the structural model's interface elements. These interface elements are not part of the CSC, they are the software data bus. Prior to execution of a typical CSC the importer element is called. The importer consumes data from other CSCs by transferring that data from a shared memory region to the CSC's local objects. After the CSC control manager runs, the software sequencer invokes the exporter element. The exporter transfers data produced by the CSC from the local objects to the shared memory region for other CSCs to import.

ISSUES

The first issue addressed is the software data bus. It must support all of the requirements imposed on the B2-ATD. The data bus must not interfere with the CSC software. It must copy data from one CSC to another and allow those CSCs to be independent of each other.

Since the simulator has many processors, the data bus requires intermediate data storage in a shared memory region. The structure of this data storage area is very important. If the data is placed in one data package for the entire simulator, Ada requires that the entire data bus be recompiled each time an interface is changed. On the other hand, interface data could be packaged one object per Ada package. This minimizes recompilation but imposes a pair of problems. The first problem is one of providing Configuration Management (CM) for all of these Ada units. The second problem is that many Ada compilers have limitations as to the amount of withing that an Ada unit can have. One of the ATD's export elements has over one thousand interfaces associated with it. At one object per data package, the export element would require over one thousand with statements which would break most compilers.

The software data structure should also allow the interface elements to move data at maximum speed. It must also provide maximum utilization of memory when storing data in shared memory.

In a single thread process the execution of any given software component can be precisely predicted with respect to the execution of all other software components. Therefore, the availability and integrity of interface data can be guaranteed. The issues of data consistency and integrity in such an environment is not a problem. Since the B2-ATD is a multi-thread, multi-rate environment, these issues are a major concern. An interface object can be updated by the producer at the same time a consumer is trying to read that data, resulting in the loss of data consistency and integrity. The software data bus must be built so that this does not happen.

Decoupling is a major issue stemming from the requirements of CSC autonomy and independent testability. An engineer needs to be able to develop software with external interfaces when the external software does not yet exist. An even harder problem is how to independently test software when the external interface is being driven. Decoupling the software modules provides maximum reusability and flexibility, both desirable qualities.

The requirement to keep the ATD current with the air vehicle magnifies the issues concerning rapid interface updates. The need to provide a means to verify interfaces and provide concordance is in the forefront. How is that information provided and how is it maintained?

INTERFACE MANAGEMENT APPROACH

Managing interfaces is not just providing a solid software architecture. An entire system must be developed to support the interfaces. The Shared Memory Management (SMM) system provides that support on the B2-ATD. The SMM system consists of two parts, an Interface Management Tool (IMT) and Software Interfaces (SI). The Interface Management Tool manages a data base of interface information and the generation of the connection manager software. It supports the update of interfaces and provides reports and documentation of interfaces. The Software Interfaces subsection of Shared Memory Management is the connection manager software generated by the Interface Management Tool. It runs on the simulator and is the software data bus.

Interface Management Tool

The IMT is actually a tool set. The heart of the IMT is the Interface Update Processor (IUP). All other tools, with the exception of the Interface Design Document. (IDD)/interface report generator, support the IUP. The IMT is layered over a relational data base. The relational data base is used by the IMT to store interface information for rapid retrieval.

Interface Update Processor

The Interface Update Processor's function is to determine interface updates, update the data base and generate new connection manager software. This process is shown in Figure 3. To accomplish this task, the declaration packages with interface updates are parsed and the interface information is extracted. The information is parsed from a combination of Ada code and comments. The types of information extracted are:

- Interface label
- Source package and object, for exports
- Destination package and object, for imports
- Object type (package and type mark)
- Initial values, for exports
- A brief description, for exports

This information allows the interface software to know where the data is exported from and imported to. It is also used in creating intermediate storage and initializing that intermediate object. The intermediate storage is discussed in the Software Interfaces section below. The interface description provides information used to document the interface. The interface label is a name given to a specific interface. It is used to make the connection between the export of the data and all imports.

The interface information need not be placed in an object declaration package. The IUP allows the information to be placed in a text file. This text file can contain the interface information for one or many declaration packages. This option is available on the B2-ATD, but is not currently being used.

The interface information found in Figure 4 is an example of information found in a declarations package. Figure 5 shows the same information as it looks in a text file. In both figures the upper case text indicates key words used to parse the files. The lower case text indicates information being extracted from the file.

In Figure 5 the initialization information for the time of day object is too long to fit on one line. The IUP allows the initialization data to be on more than one line. The ability to provide more than one line of input allows initialization of large composite types such as records of arrays of records.

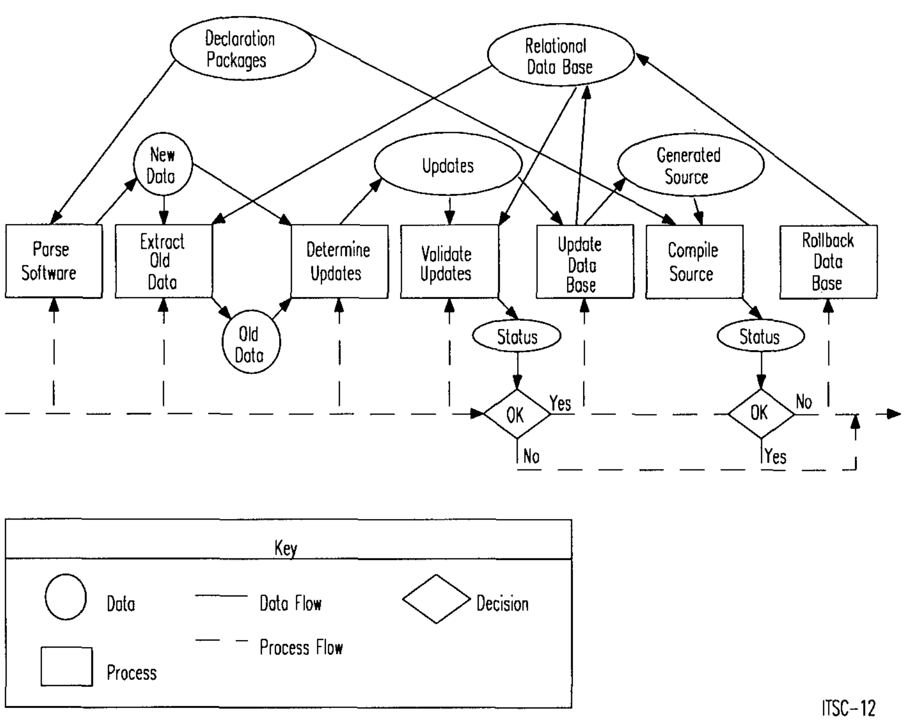


Figure 3 The Interface Update Processor

```
-->EXPORTS
-->LABEL time_of_day
   current_time : time_types.times
      := (hour=>12,minute=>0,second=>0);
-->DESCRIPTION current time of day
-->LABEL day_of_year
   current_day : time_types.days := 1;
-->DESCRIPTION current day of year
-->END
```

Figure 4 Interface Information In Source Code

```
-->PACKAGE some_data_package
-->EXPORTS
-->LABEL time_of_day
-->OBJECT current_time
-->TYPE time_types.times
-->INITIALIZATION (hour=>12,
-->INITIALIZATION minute=>0,
-->INITIALIZATION second=>0)
-->DESCRIPTION current time of day
-->LABEL day_of_year
-->OBJECT current_day
-->TYPE time_types.days
-->INITIALIZATION 1
-->DESCRIPTION current day of year
-->END
```

Figure 5 Interface Information In Text File

Next the IUP extracts interface information from the relational data base for the updated packages. With the two sets of data, old and new, the IUP determines what interface updates have been made. The updates are validated to insure that this change does not contain any detectable errors. These validation checks include the comparison of the types on both sides of the interface, insuring that unique fields of the data base are not violated and all required information is provided. It is at this time that project related rules are enforced. An example is that no export object can be deleted unless there are no imports of that data. The intent is to prevent the removal of an interface that is still required by another CSC.

Once the interface updates are validated, the relational data base is updated. With these updates in the data base, the IUP generates all import and export elements affected by this update. The newly generated data bus software and the updated declaration packages, along with any other units required for the update, are compiled against the project libraries to insure Ada compilation correctness.

In the event that the data base update has problems or the compilation check fails, the interface updates are rolled back out of the relational data base. By doing this, the IMT guarantees that the interface updates will work with the rest of the project software, at least to the point where a new executable can be linked and tested. It also insures that the relational data base is current with the project libraries.

The IUP has a software switch that can be used to generate data bus software. When this switch is used the IUP alters its process flow. The resulting process flow skips the "Compile Source" block show in Figure 3 and always passes through the "Rollback Data Base" block. This option gives application engineers the ability to do host environment testing without updating the relational data base or project libraries.

Support Tools

As stated previously, the IUP has a set of tools supporting it. The first of these tools is an interface information syntax checker. Its purpose is to insure that the interface data can be parsed from the updated units prior to submitting them to the IUP. Once an interface update is made, the software is passed through the syntax checker. If a syntax error is found in the interface information, an error message is displayed on the screen along with the line number of the error.

The second support tool is an interactive relational data base query tool. It gives application engineers access to current interface definitions. It not only allows an engineer to query the data base but it also allows the generation of custom interface reports. These reports are generated based on any or all of the data base fields. This information can be used to modifying interfaces and related software.

The information in the declaration packages is the real interface information data base. The relational data base is only used for rapid retrieval during the update, verification, software generation and report generation. In this way, standard CM tools can be used to CM the software and therefore the interfaces. With that in mind, a mass insert tool allows quick population of the relational data base. In the event a relational data base becomes corrupted, the mass insert tool can repopulate it from the CM'ed source.

The final tool is an interface definition report generator. The output of this tool is used to build a project IDD. The IDD is used for compliance with DOD-STD 2167A. It is also used by application and project engineers when designing major and minor software upgrades.

Software Interfaces

In developing the Software Interfaces, it is necessary to meet the requirement for data consistency and integrity while supporting flexibility. These constraints led to the design of the exporter and importer elements.

The exporter element is comprised of two subsections, a global memory package and an exporter procedure. The global memory package contains all of the data to be placed into shared memory, available to all CSCs. This package also contains control data which defines the state of the exporter. Figure 6 shows the structure of an exporter's global data package.

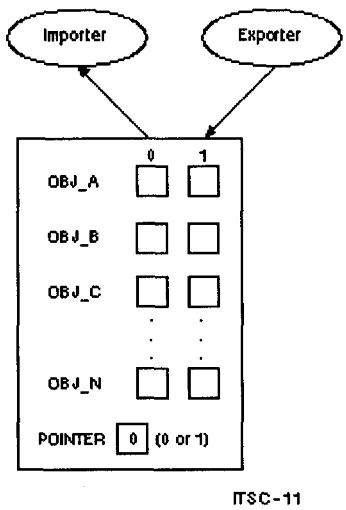


Figure 6 Data Flow In and Out of a Global Data Package

To provide data consistency, every global object in the global memory package is double buffered. One buffer is the current import object, the second buffer is the next export object. By maintaining two buffers the exporter can be writing to one buffer while an importer is reading from the other. Access to these buffers is managed through a pointer. This pointer is not an Ada access type, but an index into the buffer sets. For speed, a single pointer is used for all buffer sets.

The export procedure first determines the location of the next buffer set. It then transfers the latest data produced by the CSC into the next buffer. When completed, the export procedure updates the data pointer to point to the latest data.

The importer element contains an importer procedure. The import procedure copies the data pointers, of all exporters being imported from, to local copies. These local copies are used to retrieve consistent sets of import data. The consistency is maintained even if an exporter updates its data or pointer during the importer's processing.

A feature designed into the importer procedure is a bypass condition. The use of a bypass condition allows a control manager to turn the movement of data on or off in the importer. A bypass condition can be attached to one or more interface objects. For testing purposes, the designer may group all imports coming from a specific exporter and attach one bypass condition to this group. If the exporting CSC is not currently in the system, the importer can bypass all imports from that CSC. The interface objects can then be set to suitable default values. In addition, an emulator or an off-line development tool can be used to produce dynamic input values for the CSC. This can be accomplished without modifying a single line of code in the CSC.

The importer and exporter are structured to optimize their run-time characteristics. The emphasis is placed on speed with space as the secondary concern and lines of code the lowest priority. Development environment constraints of recompilation and relinking are also taken into account.

The exporter procedure sequencing is straight line. As stated before, the exporter first determines which buffer to store the new data in. Each export item is then move from local memory to the global memory package. Finally, the data pointer is updated. This code contains no branching and is already optimized for speed.

The importer procedure is more complex than the exporter procedure and should be optimal. The importer procedure is segmented based on bypass conditions to limit the number of "if" statements in the code. All objects without an associated bypass condition are likewise grouped together. With cache in mind, imports inside the "if" statements are grouped on an exporter boundary.

BENEFITS

In the area of testability, the architecture allows total CSC autonomy. With a local copy of the interface data, engineers can independently test their software. In the early phase of development a test driver can be developed to drive the inputs. As the overall system matures and other CSCs are added, the import bypasses can be used to turn off inputs from systems that are not yet available. Default data or a scaled back version of the test driver can be used to provide input for these interfaces.

These features also lead to simplifying the job of keeping the simulator current with the air vehicle. The local copy of interface data and the import bypasses allow a CSC to be modified to accept inputs or provide outputs to a new CSC. This new CSC may not be in the simulation software set. The existing CSC can be updated and the test driver used for regression testing. When the new CSC is added, the bypasses can be turned off allowing the data to be imported.

The interface data query tool provides valuable information in making both small and large updates to the simulation software. By using the relational data base, detailed reports can be generated to show the interaction of CSCs. This information can be used for load balancing, CSC upgrades and other similar enhancements. These reports also fill the requirement of providing a DOD-STD 2167A IDD.

The IUP provides many benefits. It allows an application engineer to update an interface by modifying a CSC's declaration package. The engineer then runs the IUP which validates the updates and generates the changes to the connection managers. This process allows easy, rapid updates to interfaces. The IUP takes a process that once took two or three days and reduces it to minutes.

By using the IUP a uniform method of moving data between CSCs has been obtained. The method of moving data can be easily changed across the simulator. A compiler upgrade may cause a rethinking of the interface software structure. A change of global data structure or the grouping of data by type may lead to execution optimization. The B2-ATD currently has over two hundred connection managers. To make a simple change to the design of the data bus software by hand would require a huge amount of time. Changing the data bus software generator takes man-days, not man-weeks.

The primary concern is data consistency. The use of double buffering and a single data pointer for an exporter has achieved that. This method allows the exporter to update the data at the same time as an importer is importing. Since the importer copies the pointer to a local storage location, the exporter can update the pointer during the importers execution.

CONCLUSIONS

As projects grow in size and complexity, the need to start managing interfaces at the earliest possible time is paramount. Interface management can no longer be viewed as a Hardware Software Integration (HSI) activity.

A software tool to automate interface management is a necessity. This tool must be able to do type checking, interface verification, interface concordance and provide a myriad of reports. In order to insure the integrity of the interfaces, this tool must have the ability to generate the interface connection software. It must also be able to generate a project level IDD. The tools for managing the interfaces must be easy to use and be reliable.

The interaction of the software architecture and the interface methodology must be orchestrated. The two need to play together to provide a software environment that is conducive to software development. An engineer must be able to rapidly affect an update to the software. That update must be of the highest quality and reliability. Only then can the claim be made that the software is truly maintainable.

While rapid updates and maintainability are required conditions, the interface software must accomplish its job. It must maintain data consistency and must do so in an optimal manner.

The SMM system allows an application engineer to alter interfaces by changing information in their software. An engineer simply tags an object with an interface label and thereby makes a connection to the producer or consumer of that data. This is done without knowing where the data comes from or where it is going to.

REFERENCES

1 "Structural Modeling White Paper" Software Engineering Institute Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, March 1992 draft

2 "Military Standard Defense System Software Development", DOD-STD-2167A 29FEB88

3 "Structural Modeling Guidebook" Software Engineering Institute Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 January 1993 draft

4 "A Graphical Notation For Software", William S. Bennett, 1991, Marcel Dekker, Inc., New York, Basel, Hong Kong

Application of Multi-Media Technology to Training for Knowledge-Rich Systems

Janis A. Cannon-Bowers
Eduardo Salas
Naval Air Warfare Center
Orlando, Florida

Phillip Duncan
Search Technology
Norcross, Georgia

CAPT E. J. Halley, Jr., USN
OPNAV (N912)
Washington, DC

ABSTRACT

Tactical decision making (TDM) can be defined as a process whereby an individual must gather, process, integrate and assimilate information in order to choose or develop a course of action that will lead to attainment of tactical goals. In order to support this process, tactical knowledge must be cognitively accessible to tactical decision makers so that they are able to recall and apply it in crucial situations. At present, the bulk of tactical knowledge is presented initially to surface warfare tactical decision makers in print format (e.g., tactical memoranda, tactical notes, and other publications). However, recent research into decision making in complex environments has shed light on the manner in which expert decision makers use knowledge in support of a decision, suggesting alternative strategies for presenting tactical knowledge in the learning process so that it is easier for tactical decision makers to remember and apply in required situations.

The purpose of this paper is to address the issue of how tactical knowledge can be presented to tactical decision makers so as to improve its retainability and useability in crucial decision making situations. To accomplish this goal, several activities were completed: (1) leveraging the work conducted under Tactical Decision Making Under Stress (TADMUS) project, a description of manner in which expert tactical decision makers employ knowledge in crucial decision making situations was formulated; (2) using this information, conclusions regarding the manner in which tactical knowledge must be initially presented to decision makers were drawn; and (3) based on the first two activities, a description of an automated system for presenting tactical knowledge that increases its retainability and accessibility in crucial decision making situations was formulated. The results of these activities are documented in this paper.

Dr. Janis A. Cannon-Bowers is a Senior Research Psychologist in the Human Systems Integration Division of the Naval Air Warfare Center. She holds an M.S. and Ph.D. in Industrial/ Organizational Psychology from the University of South Florida, Tampa, FL. Her research interests include team training and performance, crew coordination training, training effectiveness, and tactical decision making. Currently, Dr. Cannon-Bowers is principle investigator for the Tactical Decision making Under Stress (TADMUS) project, conducting research concerned with improving individual and team decision making in the Navy tactical environment.

Dr. Eduardo Salas has been a Senior Research Psychologist in the Human Systems Integration Division of the Naval Air Warfare Center since 1984. Dr. Salas also has courtesy appointments at the University of South Florida and the University of Central Florida. He earned his Ph. D. from Old Dominion University in Industrial/ Organizational Psychology in 1984. Dr. Salas is on the editorial board of Human Factors and Personnel Psychology. He has co-authored over fifty journal articles and book chapters and has co-edited four books. Dr. Salas' research interests include team training and performance, training effectiveness, tactical decision making under stress, team decision making, human performance measurement and modeling, and learning strategies for teams.

Dr. Phillip C. Duncan has been a Senior Staff Scientist at Search Technology since 1987. Dr. Duncan received his Ph. D. through a special joint program of Cognitive Psychology and Instructional Science at Brigham Young University. He received a B.S. in Psychology and a M.S. in Experimental Psychology from Brigham Young in conjunction with graduate research at Stanford University. His professional interests include team training; intelligent tutoring systems & learning environments; hypermedia training systems; human learning, problem solving, and motivation. Dr. Duncan is currently the Project Manager of a "virtual team" simulator project for the Naval Air Warfare Center.

CAPT E. J. Halley, Jr., USN, is currently Director Test & Evaluation Division on the staff of the Vice Chief of Naval Operations OPNAV (N912). He graduated from the U. S. Naval Academy in June 1967, and received a Master of Science degree in Physics at the U. S. Naval Postgraduate School, Monterey, CA, in June 1975. CAPT Halley has held a variety of positions throughout his career including: Commanding Officer of the USS Donald B. Beary (FF 1085), Branch Head for the staff of the Chief of Naval Operations in the Antisubmarine Warfare (ASW) Division (OP-71), Commander of Destroyer Squadron Thirty-One, and served as Commander, Surface Warfare Development Group.

Application of Multi-Media Technology to Training for Knowledge-Rich Systems

Janis A. Cannon-Bowers
Eduardo Salas
Naval Air Warfare Center
Orlando, Florida

Phillip Duncan
Search Technology
Norcross, Georgia

CAPT E. J. Halley, Jr., USN
OPNAV (N912)
Washington, DC

INTRODUCTION

Navy tactical decision makers operate in a knowledge-rich environment--they must remember and apply great deals of tactical knowledge in order to make crucial decisions in time-compressed situations. Briefly, this knowledge relates to the characteristics of friendly and threat assets, situational features, doctrine, applicable tactics, and rules of engagement (ROE), some of which change as a scenario unfolds. Research conducted under the Tactical Decision Making Under Stress (TADMUS) project indicates that in such situations, decision makers rely on well-established knowledge structures that are built up in memory over time. Therefore, it is essential that tactical decision makers initially encode and learn required knowledge in a manner that supports the rapid, complex decision making typical of a combat scenario.

Currently, Navy tactical decision makers acquire knowledge from a variety of sources in the training pipeline, with much of this sort of knowledge acquisition occurring on-the-job. In fact, the bulk of knowledge relating to tactics is contained in paper-based tactical publications--e.g., tactical memoranda (TACMEMOs), and tactical notes (TACNOTEs). These documents are a primary mechanism for disseminating tactical information. However, the effectiveness of these documents as a means to *teach* tactical decision makers necessary tactical knowledge is limited because the presentation of knowledge contained in them is: 1) linear in nature, 2) passive, and 3) static. At present, a number of technologies are beginning to become available that can improve training for tactical knowledge, including: multi-media reference data bases; low-cost simulation and animation capabilities; graphics presentation techniques; and advanced training technology. Moreover, data regarding the manner in which experts accomplish complex decision making tasks has led to new theories regarding how domain knowledge supports decision making; these are ripe for application to this problem.

The purpose of this paper is to address the issue of how tactical knowledge can be presented to tactical decision makers so as to improve its retainability and useability in crucial decision making situations, with emphasis on applying multi-media technology. It should be noted that the focus here is on presenting tactical knowledge during the learning or knowledge acquisition process, and not in the operational environment. That is, the idea is not to address real-time decision aiding or displays. Rather, it is to investigate how tactical decision making and tactical employment can be improved by enhancing the methods of presenting tactical knowledge to decision makers so as to optimize its comprehensibility, understandability, and ultimately, the probability that it is applied in crucial tactical decision making situations.

The remainder of this paper is structured as follows: First, the concept of tactical decision making is defined in cognitive terms, along with a description of how traditional decision making theories have viewed this process. Next, two more modern lines of thinking regarding decision making are described, and the implications of these theories for tactical decision making are offered. The reason that these theories are reviewed is because they have important implications for the manner in which tactical knowledge is best presented to decision makers during the learning process. Finally, the vision of a system for preparing multi-media, interactive, electronic tactical publications is described. Such a system would be designed to address both the effectiveness of tactical publications in supporting tactical decision making, as well as the practical and logistic issues associated with the publication process.

THE NATURE OF TACTICAL DECISION MAKING

Tactical decision making (TDM) can be defined as a process whereby an individual must gather, process, integrate and assimilate information in order to choose or develop a course of action that will lead to attainment of tactical goals. In fact, in cognitive terms, the employment of a tactic can be thought of as a complex "cue--strategy" association. That is, when confronted with a situation, a decision maker must recognize and assess information (cues) provided by the environment, and then apply an appropriate course of action (strategy, or in this case, tactic). Therefore, the task of a tactical decision maker can be characterized as 1) applying tactical knowledge in order to achieve rapid, accurate situation assessment, 2) recognizing that a particular tactic applies, and 3) taking action that is consistent with the tactic.

Research into decision making has most often focussed on situations where varying degrees of risk or uncertainty characterize the decision making event, since these factors affect the manner in which a decision is made. For many years, "classical" decision making theories assumed that expert decision makers engage in a rational, analytical process in reaching a decision under uncertainty (Beach & Lipshitz, 1993). Briefly, these theories suggested that decision makers engage in a rational process that involves selecting the optimal choice among several options by applying probability theory. In practice, these theories assumed that decision makers seek all information available to them, generate a series of viable options, assess each option based on a probabilistic determination of what they believe is likely to happen, and finally select the option that maximizes the expected outcome.

Many years of research into these classical decision theories has lead to a fairly consistent conclusion: they do not describe accurately how decision makers make decisions in the real world. There are a host of reasons why this is the case (see Klein, Orasanu, Calderwood & Zsambok, 1993). Most important to the current discussion is the fact that these theories failed to account for the context in which a decision is made. As a result, several more modern approaches to the study of decision making have evolved in recent years. These are reviewed briefly in the following sections.

Naturalistic Decision Making

An approach to studying decision making popularized recently emphasizes the importance of investigating decision makers in their natural (operational) environments (Orasanu & Connolly, 1993). Called "Naturalistic Decision Making" these theories assume that several characteristics describe the typical decision environment in 'real world' operations. According to Orasanu & Connolly (1993), these include:

- ill-structured problems
- uncertain, dynamic problems
- shifting, ill-defined, or competing goals
- action/feedback loops
- time pressure
- high stakes
- multiple players
- organizational constraints

Further, these theories suggest that decision making behavior develops over many years of experience, through exposure to many decision situations. In addition, they reject the notion that decision makers engage in a rational, analytical process in making a decision, recognizing instead that expert decision makers must sometimes make rapid decisions that necessarily short-cut the rational process. Moreover, naturalistic decision making theories assume that even the most expert decision makers can err due to situational factors, and that naturally-occurring decision biases may characterize decision making under stressful conditions.

One of the more promising naturalistic decision making theories developed to help explain expert behavior contends that decision makers employ a "recognitional" strategy in assessing a situation. That is, expert decision makers make a rapid situation assessment by recognizing patterns of cues in the environment (this process is referred to as "recognition-primed decision making"). Once the situation assessment is made, Recognition-Primed Decision (RPD) theory contends that the expert uses his/her memory of similar situations in the past to help decide on a course of action. Typically, the process of generating and evaluating options is bypassed, since expert decision makers usually know what action to take based on past experience (Klein, 1993). In a Navy tactical decision making situation, this appears to make sense--once a tactical action officer or commanding officer has made an accurate assessment of the tactical picture, ROE and doctrine often dictate (or at least delimit) what his response options may be. The result is rapid, seemingly effortless decision making, that is often characterized as "intuitive".

Evidence to support the validity of the recognition-primed decision making model for combat information center (CIC) decision making was recently found in an investigation conducted under the TADMUS program. Briefly, Klein and associates, (Klein, G., Kaempf, G. L., Wolf, S., & Thordsen, M. L., in prep. conducted interviews with 28 active duty Navy personnel from various commands (e.g., SWDG; Tactical Training Group, Pacific; Aegis Training Center and various ships). These participants represented a variety of experience levels and CIC duty stations. The interviews sought to understand more fully the processes that CIC personnel employ in making crucial decisions, and to identify critical cues that decision makers use in reaching anti-air warfare (AAW) decisions. Results of these interviews and subsequent analyses indicated that CIC decision makers appear to invoke recognition-primed processes when making most of their decisions. In fact it appears that over 90% of the situations experienced by the AAW team are either highly or moderately familiar to them. This recognition then triggers recall of many associated pieces of information, including expectancies, goals, and appropriate actions. Moreover, decision makers rarely generate and/or consider more than one response option once a situation assessment is made.

Several aspects of recognition-primed decision making theory are particularly important to the dissemination and learning of tactical knowledge. First of all, the theory suggests that expert decision makers develop a series of situation "templates" in their memory over time. These templates are generalized cases of common situations that contain knowledge about situations they have encountered (e.g., the cues that describe the situation), along with knowledge regarding the correct responses or course of action associated with that situation. When a decision maker is confronted with a new decision making situation, the theory suggests that he/she might solve it by using memories of past situations as a guideline. Of course, there may be aspects of the current situation that are novel (i.e., not in the expert's memory). In these cases, the expert must rely on knowledge of similar situations or modify the template in order to make a decision (see Klein, 1993 for a more detailed description of this process). Of importance here is the notion that exposing decision makers to numerous decision making scenarios is a useful means to build necessary situation templates.

Another feature of RPD theory worth noting is that it is particularly applicable to situations where time compression is a factor. In fact, RPD may be best at explaining decision making in situations where time pressure limits a decision maker's strategy. In these situations, decision makers appear to spend little, if any, time decomposing a situation in order to understand it. Instead, it appears that entire *patterns of cues* are perceived simultaneously by decision makers, and that the significance of each of the individual cues is some how "digested" along with other cues in the situation.

A study of chess masters supported this idea.

Briefly, it was found that expert chess players were significantly better than novices in remembering the placement of pieces on a chess board when the placement was a "legal" one--that is, one in which the pieces could have conceivably landed during the course of a game. In contrast, chess masters were no better than novices at remembering the placement of pieces when it was random (i.e., where pieces were place haphazardly with no regard for whether the placement was feasible). This study suggests that expert decision makers may, over time, encode entire patterns of situational cues. Once again, the implication here seems to be that in order to improve expert decision making, decision makers must be exposed to likely scenarios so that they can build appropriate memory structures.

Finally, RPD predicts that when a decision maker is uncertain regarding whether or not to take a course of action, he/she engages in a kind of "mental simulation" of the solution. That is, the decision maker plays out the implications of the decision in his/her mind in order to evaluate it before taking action (Klein, 1993). If this mental simulation indicates to the decision maker that the option is a viable one, then the option is exercised. If potential problems are encountered, then the decision maker will modify the course of action so as to ameliorate the problem. This concept is important because it suggests that fostering the mental simulation process may improve the quality of decision making.

Knowledge Structures

A second line of inquiry that bears on the discussion of TDM involves the study of knowledge structures or mental models. "Mental models" can be defined as dynamic cognitive representations that allow people to describe, explain and predict events in their environment (Rouse & Morris, 1986). Mental models contain organized information that describe objects, properties, causal connections and relationships in systems or situations in the environment (Cannon-Bowers, Tannenbaum, Salas & Converse, 1991). For example, a car mechanic may have a mental model of how a car's engine operates. This model (which describes how and why certain components of the engine are related) helps him/her to diagnose problems, troubleshoot, and ultimately repair car engines. Similarly, it has been suggested that Navy tacticians (particularly CIC personnel) have mental models of the tactical task and situation (Rouse, Cannon-Bowers & Salas, 1992). For example, an Aegis Anti-Air Warfare Coordinator (AAWC) may hold a model of the 1.) Aegis system containing knowledge about how the system works, its components, the rationale behind its operation, and the like; 2.) the console with which he is interacting containing proceduralized knowledge regarding how to interact with the console (which buttons to push, etc.) in order to accomplish various goals; and 3.) the more general AAW task containing knowledge about the physics of missile and platform movement, likely air threat characteristics, appropriate tactics, and so forth. Each of these models contributes to his ability to make decisions by providing an organized framework in which tactical knowledge can be cast.

It should be noted that the concept of a mental model described here is similar to the notion of a "situation template" discussed above. That is, experts appear to rely on pre-existing knowledge structures when making a tactical decision. Moreover, it is important to note that these theories contend that *tactical decision making effectiveness depends on well organized tactical knowledge*. In fact, the study of differences between expert and novice decision makers has revealed that experts have mental models that are organized around "deep" underlying principles, whereas novice models are organized around more shallow surface features (Chi, Feltovich, & Glaser, 1981). In addition, expert models seem to be more abstract, pattern-oriented and highly integrated (Cannon-Bowers et al., 1991). Taken together, these findings suggest that in order to be most effective, *expert tacticians must hold accurate and complete mental models to support the TDM process.*

TACTICAL KNOWLEDGE AND TDM

The work cited to this point has important implications for the manner in which tactical knowledge should be presented in the learning process to support comprehension and later TDM. In fact, the overriding message of this paper is that the manner in which tactical knowledge is initially presented to decision makers will have an impact on the extent to which they are able to effectively encode and use that knowledge in decision making. Before addressing the topic of information presentation directly, several issues regarding the properties of tactical knowledge itself must be discussed.

The Nature of Tactical Knowledge

Tactical knowledge can be thought of as "knowledge with a purpose". That is, the role of tactical knowledge is to support effective TDM (as opposed as simply being interesting to know). Therefore, it is useful to examine the manner in which pre-existing knowledge in a situation can foster the rapid, flexible type of TDM performance demanded by modern warfare.

Traditionally, the discussion of knowledge and decision making has centered around identification of "decision rules" in a task domain. That is, it was often assumed that knowledge was stored (or at least accessed) in the form of an "IF...THEN" rule. For example, "if the contact is coming towards me and descending, then it is probably hostile". The problem with this perspective is that it cannot accurately describe decision making in situations that are ambiguous, complex, dynamic, time pressured, or not well specified--all characteristics of many tactical decisions. Therefore, modern research has sought to examine the knowledge required for decision making in terms of *flexible* knowledge structures (or mental models) as described above (e.g., see Cannon-Bowers et al, 1993).

In addition, researchers of late have begun to distinguish among different types of knowledge that characterize decision making. These include "declarative" knowledge, which is knowledge about the facts, concepts, rules and relationships in a task area; "procedural" knowledge, which is knowledge about the steps required in performing a task or accomplishing a task; and "strategic" knowledge, which is a more complex form of knowledge that combines declarative and procedural knowledge with situational knowledge that indicates how and when to apply pertinent knowledge.

Figure 1 relates this discussion of knowledge type to the notions about mental models presented above. According to this figure, a decision maker's pre-existing mental models contain declarative and procedural knowledge about various components of the system or situation. For example, an AAWC's mental model of the Aegis system might contain facts about the system itself (e.g., how it operates, how it is related to other systems, why it works the way it does, etc.), along with procedural knowledge about how to interact with the system in order to accomplish particular tasks or goals. Knowledge in these mental models may also be more situational in nature. For example the situation model may contain facts about the geographic region in which the event is occurring (e.g., likely scenarios, appropriate responses, etc.), while the team model might contain information about the strengths and weaknesses of particular teammates (see Figure 1).

Figure 1 specifies further that these preexisting mental models contribute to TDM performance by formulating a current "problem model". This problem model includes strategic knowledge that helps the decision maker to arrive at a situation assessment by selecting from memory important declarative and procedural mental models. In other words, the problem model matches the cues (or cue patterns) perceived by decision makers in a situation with appropriate knowledge templates from memory that will help him/her to make the decision.

From a practical standpoint, the line of thinking proposed here would suggest that when an expert decision maker is confronted with a decision problem, patterns of cues are perceived that trigger the recall of particular knowledge structures (mental models) that together form the basis of a dynamic problem model (dynamic in the sense that it is continually updated by new information from the environment). This model then suggests which, if any actions apply. Going back to the question posed early in this paper; namely, how does tactical knowledge affect TDM, it is now possible to draw the following conclusions:

- TDM requires decision makers to hold accurate, complete declarative mental models regarding the systems and situations of interest

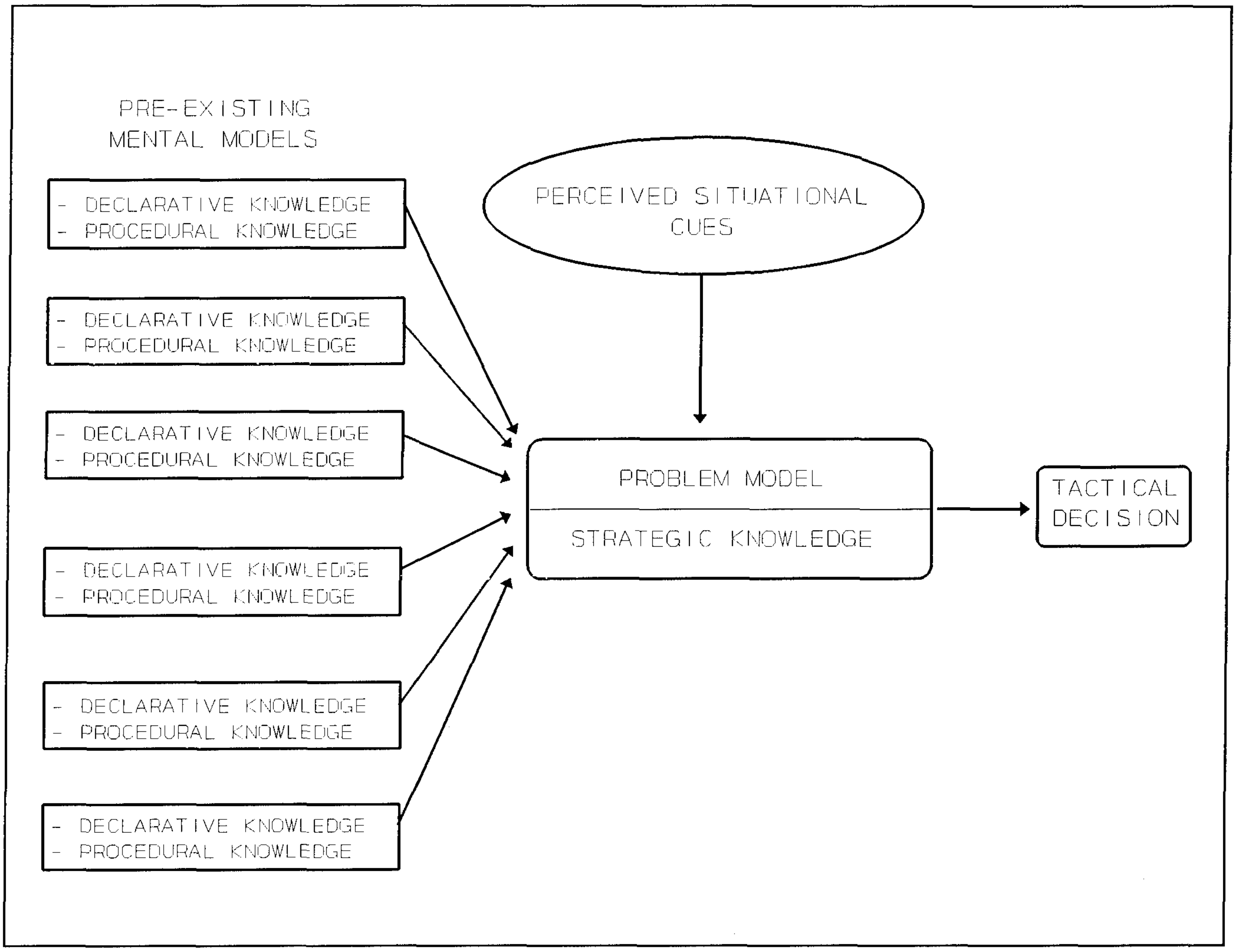


Figure 1:Model of TDM

- TDM requires decision makers to hold accurate, complete procedural mental models of the systems, tasks and routines associated with the decision situation

- TDM will be fostered to the extent that pre-existing mental models are well structured in the decision maker's memory

- TDM requires the formulation of accurate problem models through application of appropriate strategic knowledge (which is built up through experience)

Supporting Application of Tactical Knowledge

In a general sense, the notions about decision making and tactical knowledge presented thus far have important implications for improving the tactical decision maker's ability to optimally apply tactical knowledge to a TDM situation. These include:

1. Present tactical knowledge to decision makers (initially) in a format that fosters development of accurate and complete declarative, procedural and strategic mental models using guidelines specified above

2. Foster the formulation of appropriate cue--strategy associations by:

-- Identifying important situational cues that trigger particular responses
-- Creating scenarios that expose decision makers to various cues and associated TDM strategies

-- Providing ample exposure to a variety of scenarios via simulation

3. Provide a context in which tactical knowledge can be comprehended, understood, and encoded into memory

OTHER ISSUES IN TDM

Now that the cognitive requirements of tactical knowledge presentation have been established, attention can turn to other issues that must be considered in the tactical publication and employment arena. These include team performance issues and motivational issues. The following sections address these topics.

Team Issues

Recently, researchers working in the team training and performance area have extended mental model theory (see above) to incorporate the team aspects of TDM (see Cannon-Bowers et al., 1993). Briefly, the idea here is that tactical decision makers in the interdependent, hierarchically-organized CIC environment must depend on one another to provide information that is crucial to effective decision making. As such, it is assumed that team members who have common or shared mental models of the situation, task and team will be better able to support the information needs of their teammates. A sports analogy may help illustrate this point: the blind pass in basketball is an example of a situation where two team members make an assessment of a complex set of environmental cues (the physical position of players, time left on the clock, the coach's style, the ability of teammates, and the like), and then execute a *compatible* (not identical) behavioral response (i.e, one throws the ball, while the other catches it). Team researchers have labeled this type of behavior as "implicit coordination" because it is coordinated performance that occurs in the absence of explicit strategizing or communication, (Kleinman & Serfaty, 1989). In a CIC, the same type of coordinated behavior is desirable, particularly when time constraints mitigate the amount of discussion that can occur regarding a tactical situation.

The role of mental models in implicit coordination is that they provide team members with underlying knowledge required to anticipate the needs of the decision and their teammates. The implication of this contention for tactical knowledge presentation is, once again, that the manner in which tactical knowledge is initially presented to decision makers will have an impact on the compatibility of their mental models with other team members. For example, if tactical knowledge is presented such that multiple team members can encode the information (i.e., make correct cue--strategy associations), discuss its implications (e.g., talk out the possible scenarios in which the tactic might apply), share their understanding of the tactic (e.g., why it might work, its limitations), and practice employing it, then team tactical decision making should benefit. Therefore, attempts to improve the process of presenting tactical knowledge should allow for all relevant team members to become involved in learning the tactic initially.

Motivation

It has been well established in the research literature that the learning and retention of information is affected greatly by the learner's motivation (Tannenbaum, Cannon-Bowers, Salas & Mathieu, 1992). In fact, motivation to learn and to transfer knowledge and skill back to the job are both potent factors in determining training effectiveness. With respect to the current discussion, it is clear from interviews and observation that motivation to read and retain tactical knowledge in print format could be improved. A viable way to do this is to host tactical knowledge in a manner that is more engaging to users. More importantly, it has been found that making the learning process "active" (i.e., letting learners asks questions, simulate decisions, etc.) is more effective than "passive" learning. Once again, this fact supports the notion that an interactive, multi-media presentation of tactical knowledge might be more effective than current hard-copy presentation.

APPLYING MULTI-MEDIA TECHNOLOGY TO TACTICAL KNOWLEDGE

As has been stated, modern computing technology provides a basis to improve the presentation of tactical knowledge. Specifically, a system can be conceived that will not only address the cognitive issues raised above, but also reduce the cost to produce tactical publications, and reduce the Navy's dependence on contractors to write them. Such a system could benefit from the following technologies: referenced data bases, advanced graphics presentation, cognitive engineering, user authoring tools, animation/simulation, performance measurement, knowledge organization, and multi-media presentation formats. By way of review, such a system could be developed expressly to accomplish the following goals:

- enhance the comprehensibility and useability of tactical knowledge

-increase the accessibility of tactical knowledge on board ship

- enhance the training/learning value of tactical publications

- increase the users' motivation to learn tactical knowledge

- improve the team's ability to coordinate TDM and tactics employment

- decrease the Navy's dependence on contractor support

- decrease the cost of tactical publications

- ultimately, enhance the quality of TDM and tactics employment

To accomplish these goals, a PC-based, multi-media system was conceived that could provide the decision maker with organized knowledge about the tactic and tactical situation in which it applies. Based on work conducted under TADMUS, this knowledge would be categorized into six related knowledge bases (situation/ship, tactics, team, system, task, equipment) that form the basis of mental model development. Knowledge in these areas would be classified further as supporting "why", "how" and "what" questions regarding the tactic. For example, for a particular tactic, the team model would contain information on *what* role each of the team members play in employing the tactic, *why* various team members are involved in the tactic (and associated TDM), and *how* team members interact to achieve the tactical objectives.

The vision of how such a system might be used is as follows: Beginning with the subject matter expert, the system would guide him/her through a process of extracting the tactical knowledge that underlies TDM. This process would proceed in a structured manner, organized around the knowledge categories listed above, and guided by a series of probe questions. The overriding goal of this procedure would be to describe accurately the various situations for which the tactic applies, identify and make salient the important cues in the situation that should trigger employment of the tactic, explain in detail how all pertinent equipment and systems are involved in the tactic, and delineate the role of all team members involved in the tactic employment and associated TDM task. For example, a tactical publication may contain information regarding which cues in the *situation* will trigger use of the tactic (particularly deviations from "normal") and why these cues are significant, which *ship* systems will be affected by the tactic and how these will be affected, which *team* members are involved in the tactic and their particular roles, which *systems* are affected/employed by the tactic and how these are affected, which *task* procedures are appropriate to accomplish the tactic, and how the *equipment* needs to be configured and why it needs to be configured this way.

Once developed, the tactical publication would be sent to the ship via computer media. As conceived, the this system would provide ample flexibility in how knowledge is presented to the user. To begin with, different levels of knowledge would be available so that users could tailor the information they receive to meet their specific needs. Second, the system would provide the knowledge required to build necessary mental models in order to support TDM. Third, the system would be useful for multiple players, since tactical knowledge could be presented to a team simultaneously, with intermittent discussion (to foster compatible knowledge structures). Forth, the system would incorporate multiple media presentation capabilities (e.g., video, audio, simulation) so that cues from different modalities could be represented. This feature would also help ensure that motivation to understand the material was high. Fifth, the system would allow users to access information in a sequence and at a pace that is comfortable to them.

Finally, a simulation capability would be incorporated into the system that would allow users to simulate situations to see and hear how a scenario requiring the tactic might unfold. This feature could be developed expressly to ensure that all important situational cues are made salient, and used as a mechanism to provide the decision maker with exposure to pertinent scenarios. Going back to the discussion of decision making, this would help to build crucial mental models, and foster rapid recognition-primed decision making. This capability could also allow decision makers to pose "what if" questions to see the implication of various decisions on the outcome, thereby aiding their ability to mentally simulate the implications of a decision. In addition, by allowing team members to "see the world" from the perspective of their teammates, the building of common mental models would be enhanced. This would be accomplished by providing team members with the ability to experience scenarios from various CIC positions.

With respect to cognitive functioning, the system described here would meet the four requirements noted earlier. Namely, it would present knowledge to decision makers in a format that fosters development of accurate, complete mental models; it would help to identify necessary declarative and procedural knowledge required to support tactics employment, it would foster formation of appropriate cue--strategy associations via simulation (i.e, exposure to scenarios), and it would provide a context in which tactical knowledge can be understood and encoded into memory. Moreover, the described system meets the more general objective delineated above. Specifically, it could help to enhance the useability of tactical knowledge, increase the accessibility of tactical knowledge on board ship, enhance the training/learning value of tactical publications, increase the users' motivation to learn tactical knowledge, improve the team's ability to coordinate TDM and tactics employment, decrease the Navy's dependence on contractor support, and decrease the cost of tactical publications. Ultimately, such a system could enhance the quality of TDM and tactics employment, and improve the efficiency of the tactical publication process as well.

REFERENCES

Beach, L.R., & Lipshitz, R. (1993). Why classical decision theory is an inappropriate standard for evaluating and aiding most human decision making. In G.A. Klein, J. Orasanu, R. Calderwood, & C.e. Zsambok (Eds.) Decision making in action: Models and methods (pp. 21-35). Norwood, NJ: Ablex.

Cannon-Bowers, J.A., Salas, E., & Converse, S.A. (1993). Shared mental models in expert team decision making. In J. Castellan Jr. (Ed.), Current issues in individual and group decision making (pp. 221-246). Hillsdale, NJ: Tannenbaum, Lawrence Erlbaum.

Cannon-Bowers, J.A., Salas, E., Tannenbaum, S.I., & Converse, S.A. (1991). Toward an integration of training theory and practice. Human Factors, 33, 281-292.

Chi, M.T.H., Feltovich, P.J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. Cognitive Science, 5, 121-152.

Klein, G., Kaempf, G. L., Wolf, S., & Thordsen, M. L. (in prep.). Decision-making in the AEGIS Combat Information Center. Naval Command, Control, and Ocean Surveillance Center Technical Report. San Diego, CA.

Klein, G.A. (1993). A recognition-primed decision (RPD) model of rapid decision making. In G.A. Klein, J. Orasanu, R. Calderwood, & C.E. Zsambok (Eds.) Decision making in action: Models and Methods (pp. 138-147) Norwood, NJ: Ablex.

Klein, G.A., Orasanu, J. Calderwood, R., & Zsambok, C.E. (Eds.) (1993). Decision making in action: Models and methods. Norwood, NJ: Ablex.

Kleinman, D. L., & Serfaty, D. (1989). Team performance assessment in distributed decision-making. In R. Gilson, J. P. Kincaid, & B. Goldiez (Eds.), Proceeding: Interactive Networked Simulation for Training Conference, Orlando, FL.

Orasanu, J. & Connolly, T. (1993). The re-invention of decision making. In G.A. Klein, J. Orasanu, R. Calderwood, & C.E. Zsambok (Eds.) Decision making in Action: Models and Methods (pp. 3-20). Norwood, NJ: Ablex.

Rouse, W.B., & Morris, N.M. (1986). On looking into the black box: Prospects and limits in the search for mental models. Psychological Bulletin, 100, 349-363.

Rouse, W.B., Cannon-Bowers, J.A., & Salas, E. (1992). The role of mental models in team performance in complex systems. IEEE Transactions on Systems, Man, and Cybernetics, 22, 1296-1308.

Tannenbaum, S.I., Cannon-Bowers, J.A., Salas, E. , & Mathieu, J.E. (1992). Deriving theoretically-based principles of training effectiveness to optimize training system design. Proceedings of the 14th Annual Interservice/Industry Training Systems Conference (pp. 619-631). Washington, DC: National Security Industrial Association.

TRAINING EXERCISE PLANNING: LEVERAGING TECHNOLOGIES AND DATA

Dr. Mona Crissey, MAJ George Stone, CPT David Briggs
Simulation, Training and Instrumentation Command
Orlando, FL
Dr. Mansooreh Mollaghasemi
University of Central Florida
Orlando, Fl

Recognizing that future battlefield training and preparation for "other than war" missions will rely more and more on simulators and simulations, unit commanders must incorporate new ways to efficiently use their limited resources to develop effective training plans. Currently, commanders spend hours referring to training and field manuals, training records, unit standard operating procedures and directives to develop how best to train under resource-declining conditions and limited training opportunities. Innovative methodologies must be applied to the planning process to match essential task lists against proper training resources. Also, assessments of previous training events must be fully integrated into the planning process to ensure a unit learns, and returns to train at a higher state of readiness. This paper describes a technology demonstration program being developed by Simulation, Training and Instrumentation Command (STRICOM) called Combined Arms Tactical Trainer Training Exercise Development System (CATT TREDS). The system will provide unit commanders with an intelligent decision support tool to save planning time, enhance unit training options, and automatically apply after-action review feedback in a process applicable to planning maneuver, combat support, and combat service support training, as well as, military operations other than war exercises. Some state-of-the-art technologies such as expert systems, multi-criteria decision making, voice recognition, and neural networks have been investigated for their use, adaptability, and applicability for the tool. Commercial off-the-shelf (COTS) software packages with capabilities to link applications in an object-oriented, intuitively user-friendly manner have been evaluated. Leveraging capabilities inherent in these technologies, software packages, and previously developed databases shows great promise for development of a tool allowing unit commanders to optimize training exercise planning time.

Dr. Mona Crissey is the CATT TREDS Project Director for the Project Manager Combined Arms Tactical Trainer (PM-CATT) at STRICOM, 12350 Research Parkway, Orlando, FL 32826-3276, (407) 384-3242. Dr. Crissey holds an EdD in Education from University of Alabama and an MA in English from University of Kentucky. She has 14 years of program and database management experience with both industry and government.

MAJ George Stone is the project engineer for CATT TREDS. He is a graduate of West Point and holds an MS in Industrial Engineering from Texas A&M. He is currently in the Industrial Engineering doctoral program at University of Central Florida. He is a former Assistant Professor in the Systems Engineering Department at West Point, and has conducted research in the areas of combat development and training systems.

Dr. Mansooreh Mollaghasemi is an Assistant Professor in the Department of Industrial Engineering and Management Sciences at University of Central Florida and specializes in simulation systems, operations research and multi-criteria decision making. She holds a Ph.D. in Industrial Engineering from the University of Louisville and holds other degrees in chemical engineering.

CPT David Briggs is the Assistant Project Engineer for CATT TREDS. CPT Briggs holds an MBA from Boston University. He is currently in the Master's degree program for Operations Research and Simulation at University of Central Florida. He has held field artillery command and staff assignments to include serving in Desert Storm with the 82nd Airborne Division.

TRAINING EXERCISE PLANNING: LEVERAGING TECHNOLOGIES AND DATA

Dr. Mona Crissey, MAJ George Stone, CPT David Briggs
Simulation, Training and Instrumentation Command
Orlando, FL
Dr. Mansooreh Mollaghasemi
University of Central Florida
Orlando, Fl

BACKGROUND

Facing a world filled with multiple global missions and increased task diversity, the military continues to invest in a virtual simulation-based training environment to offset drastic cuts in training areas, units, and personnel. Preparation for virtual training requires extensive planning and may incorporate a variety of static and dynamic media forms such as text, data, graphics, still images, animation, full-motion video, speech and nonspeech audio. (Ragusa, 1994). Planning tools being developed to meet the training challenges of the future should consider using some of these available technologies and leveraging available data from databases already generated.

Future Training Requirements

The training environment of the future must be organized to meet many new world realities. Military operations other than war including peace-keeping, peace enforcement, and humanitarian assistance must be addressed in addition to conventional training scenarios. The increasing use of joint, combined, interagency, and international forces requires a closer look at overlapping task responsibilities. Digitization of the battlefield requires compatibility of not only weapon and communications systems, but training systems as well. The major challenge facing the training community is to effectively train for these new and increased requirements with a downsized military population, reduced budgetary resources, and severely restricted field training areas.

Examining Training Strategies In A Distributed Interactive Simulation (DIS) Environment

With increasing constraints and environmental limitations on traditional field training, there is a definite move toward, and a major reliance on, a simulation-based training strategy. Simulating real world situations using mathematics, computers, and symbols or icons in training devices is a common occurrence today. Simulations leverage many military areas, allowing training for dangerous combat situations or testing of future equipment without endangering the environment or equipment, while protecting the safety of personnel. DIS technology allows large numbers of simulated systems, both manned and unmanned, at different locations to interact at the same time to accomplish a common training mission via communication networks. Tasks to be trained, equipment and personnel resources to be used, and the degree of difficulty of exercise conditions must all be considered. Effective planning is essential for optimal use of varying conditions, personnel and equipment resources, whether real or simulated, while still adhering to training mission standards.

Using Developed Technologies To Solve Planning Problems

Structured planning and preparation of a training exercise can be very time consuming. An orderly process is warranted that takes into consideration the commander's guidance, training doctrine, scheduling needs, and unit proficiency assessments. Mission analysis, course of action development, and event synchronization and execution matrix building all require time to accomplish. There is no shortage of automated tools and technologies designed to save time, give intelligent user help, and provide guidance for decision making for other areas. To help solve many planning problems, existing technologies and capabilities provide a wealth of resources to examine for applicability to exercise planning. Automated schedulers, expert systems, multi-criteria decision-making support aids, neural networks. relational database structures, automated spreadsheets, and object-oriented database development are only a few of the possibilities.

Other Software Development Issues

Other issues involved in development of software for military use must also be considered. One such issue is the time required for complete development and implementation of the application package. Another is the fact that most software packages are designed to run on UNIX-based platforms, not on the personal computers (PCs) with which many users are familiar. The use of Windows as an operating environment is quickly becoming a necessity when developing interactive programs. In viewing the threat of Windows NT technology on the UNIX architecture, the CEO and President of Intel, Andy Grove, predicts that the only way to succeed in new program development is to be compatible with Windows (Uninews, 1994). Therefore, if the goals for a new system development are to rapidly demonstrate a tool for planning training exercises, incorporate changes in a timely manner, and field a working model quickly, it would seem that those goals could most effectively be achieved via a Windows-based PC platform of commercial off-the-shelf (COTS) programs and previously developed databases.

CONCEPT FOR CATT TREDS

When given the opportunity to train in a simulator-based exercise environment such as SIMNET (SIMulator NETworks), a unit commander must first prepare the unit for training. The operations order must be prepared, the exercise planned, and direction to the unit must be provided. The ultimate goal of the planned exercise is to optimize the training benefit for the time, manpower, and costs involved, The success of the plan can be measured by the increased proficiency of the unit after training. The current military training development planning process requires the commander to work fast under archaic semi-automated conditions. The manual manipulation of reports, files, operations documents, maps and overlays is tedious and time-consuming. Unfortunately, the amount of preparation time never seems adequate. Lack of planning time can result in insufficient and ineffective training exercise plans, battle scenarios, and use of training time. The training exercise may become a "hit or miss" opportunity to improve specific training deficiencies. Recognizing these difficulties, the Combined Arms Tactical Trainer Training Exercise Development System (CATT TREDS) presented an automated solution to the time constraints and requirements faced by a company-level commander while preparing plans for training exercises.

Why Is There A Need For A CATT TREDS?

A need existed for a tool to help streamline the training exercise planning process. The purpose of such a tool was to provide an automated, intuitively user friendly system to minimize planning time and maximize the available training time. One of the primary drawbacks to development of a tool meeting those requirements has been the lack of computer programs which can automate tasks and actions fast enough for commanders and their training staffs to react in an accelerated mode and still plan meaningful exercises. Saving time while automating the training exercise planning process in a faster more efficient way is the goal of the CATT TREDS tool.

Who Is The Intended User?

Currently, the target user audience is the training personnel in a battalion who are responsible for platoon, company and battalion-level training exercises and planning. The primary focus of the battalion staff is on integrating platoon and company training with that of the battalion and higher echelons and ensuring that training plans coincide and complement each other. It is expected that company commanders could use CATT TREDS to plan their unit's training exercises with minimal computer experience and system training time.

TECHNOLOGIES CONSIDERED FOR INTEGRATION

The following technologies are being considered for integration into the CATT TREDS tool. Consideration is based on the ease of use, possibilities for seamless integration, and applicability to the function to be accomplished.

Knowledge-Based Expert Systems

Expert systems technology appears to be an appropriate means to employ when the following conditions occur:

- The problem at hand cannot be effectively solved with conventional programming.
- The integration of an expert system with multi-media offers the potential to improve

advisory, training, education and presentations applications from large data repositories.

According to El-Najdawi and Stylianou, (1993) "Expert systems are computer programs that incorporate the knowledge of one or more human experts in a narrow problem domain and can solve problems that the expert(s) ordinarily solve." Some benefits to be expected when using expert system technologies "include:

- Ability to capture critical expertise
- Faster application development
- Ability to distribute knowledge
- Flexibility to free experts from making repetitive decisions
- Ability to combine knowledge from several experts" (El-Najdawi and Stylianou, 1993).

Exercise Options. In developing a training exercise scenario, the unit commander must specify the criteria for success. Foremost in this development process is the issue of knowledge acquisition which sets limits and bounds. Experienced commanders draw on their own knowledge and past experience to develop scenarios that they know will meet with success and incorporate accepted doctrine and strategies. Building on this capability, well-accepted rules map an expert's description of scenarios that meet selected criteria to solve the problem.

The expert system being developed for CATT TREDS applies rules and conditions that are based upon previous experiences of other commanders to provide options that meet the selected criteria. The embedded knowledge-based expert system allows the transfer of knowledge to the commander in real time. The scenario(s) suggested may be used with confidence that they effectively solve the training "problem" and will successfully meet accepted standards.

Event Feedback. Feedback is needed by the unit as soon as a training event is completed. An After action review (AAR) may employ any number of multimedia tools and reports to draw an assessment picture for the unit commander. Usually the experts or observer-controllers who collect AAR information must recall a myriad of data and details for a ten minute briefing. The use of an expert system as an aid in this information capture process could ensure that all key teaching points, major events, and learning objectives are met. The expert knowledge base, in this case, could collect vast amounts of information during the training session, sort it quickly, and provide succinct recall directly after the exercise.

Neural Networks

"A neural network is a dense interconnection of computationally simple processors (i.e., neurons) that is based on the anatomy of the brain. Neural networks do not allow us to solve computational problems that have been unsolvable in the past. They simply provide a different way of solving a problem that may or may not lead to a better solution than some alternative method." (Georgiopoulos and Heilman, 1993).

Logistical Information Integration. The integration of logistical information into the training exercise scenario appeared to be a place to investigate the use neural networks. In this case a neural network could replicate a commander's thought process in building the criteria for equipment and personnel resources. Morrison (1992) suggested that "in non-lexical problem solving domains, the patterns applied by experts to classify their environmental stimuli and the mental models from which they generate responses, incorporate spatio-temporal patterns that can not be implemented under the current symbolic paradigm." In other words, the commander's thought process may be too complex and unknown for adequate modeling via conventional techniques currently in use. With software applications and hardware for neural networks enabling dynamic transfer of data between routines, the neural network submodel presents a viable option for representing temporal and spatial relationships, especially applicable to logistical determinations.

Training Paradigms Automated. Another possible use for neural network technology is in the area of collective tasks. Training in the Army permeates through initial basic individual skills to larger unit or collective abilities. From this perspective, training captures the essence of learning layers of tasks. Once lower level or individual tasks are learned, an automated neural network system could then model this level perfectly, omitting errors to work solely on the tasks at the next level. A neural network of trained individual tasks can track the unit's training as a collective team. This integration of multi-task, multi-echelon training allows the commander to focus his efforts on collective training, while maintaining the unit's individual training tasks.

Voice Recognition

"Computer users have always yelled at their machines. But now the computers are beginning to listen." (Thyfault, 1994) In looking at technologies that would enhance its user friendly capabilities, voice recognition technology was a natural for evaluation for use in CATT TREDS. Using a digital signal processor installed on an 80486/50mhz computer, IBM has made a speech server which allows good dialogue with a PC after several iterations of training to recognize user accent and pronunciation style (Andrews, 1993).

For CATT TREDS, several voice recognition packages with similar characteristics are being tested on a multimedia personal computer. Even though these systems recognize voice commands at an affordable price, the downside is that usually several hours of time are needed for "training" the system to respond to user commands. Typical response reliability has not been consistent during initial research and trial efforts. One inexpensive system tested with a 40 word vocabulary provided a 91% response accuracy when trained and used by the same individual, but only a 57% response accuracy when trained for generic voice and used by an individual.

In addition, recent testing on the Toshiba multimedia system which uses DragonWriter software revealed a longer response time using voice commands than using the mouse to execute the same command. For the experienced mouse user, this system might seem much slower and inefficient, thus defeating the purpose of incorporating voice technology. More research is required to find an affordable tool with high response accuracy that does not require individual training.

Multi-Criteria Decision Making (MCDM)

Almost all real life decision problems involve multiple objectives. There is really nothing new about multiple objective problems. Humans have been making such decisions throughout history. These problems have often been resolved through the use of intuition or by various processes of choice that developed over time. Simple problems (i.e., those involving a few objectives and a small number of alternatives) can usually be solved without the use of sophisticated methods. Only when the number of objectives and alternatives increase does the need for formal techniques become acute. In the presence of a large number of conflicting objectives and numerous alternatives, the use of techniques that aid the decision maker in structuring his preferences and criteria is necessary. This is due to the difficulty encountered in articulating tradeoff information and maintaining the required consistency. In choosing a solution, the decision maker must be willing to accept a loss in one or more of the objectives, or tradeoff one objective in order to increase the value of another objective. This tradeoff information is often very difficult to elicit especially in the presence of a large number of criteria. With the use of more formalized techniques, the decision maker is guided throughout the process, and can reduce the cognitive burden and ensure consistency.

Over the past 20 years there has been a plethora of tools and techniques developed for solving multiple criteria problems. These multi-criteria decision making methods are designed to clarify the decision problem, help generate useful alternative solutions, and help evaluate the alternatives based on the stated preferences. They generally involve the use of computer models.

MCDM Application To CATT TREDS

In the application of MCDM to CATT TREDS, we envision using a ranking system to identify the best scenario for training. Given the tasks to be trained, the applicable METT-T factors (i.e., mission, enemy composition, time available, terrain, and troops), and the desired situational training exercise (STX) chosen, an expert system will identify several scenarios from a scenario library that fit the selected objectives of the training exercise. To arrive at a preferred scenario, the commander identifies mandatory tasks and rank orders the tasks to be trained. A matrix is then generated that identifies how well each Course of Action (COA) in the scenario library meets the training requirement of each task. The commander may then select a scenario that, in his judgment, best allows the unit to conduct the desired training.

Intelligent Multimedia Applications (Intellimedia)

Intelligent Assistance. Using 110 colorful lights, 36 water pumps, a stereo system and 1,000 feet of hose, Sea World built a magnificent water fountain show that held young and old alike spellbound for over twenty minutes. Each object used was a common, household item that creates little interest by itself when used to water a lawn, provide music or light a Christmas tree. Likewise, applications of media integration and synchronization aided by technology have been used to create captivating and entertaining courseware, tutorials and textbooks. While expert systems enable users to draw information from a large database, multimedia features provide graphical and realistic representation of information for making training decisions (Ragusa, 1994). As noted by Marchionini and Crane (1994), this process of integrating multimedia with a learning process such as setting up a training plan is not easy. Further research is still required to define workable goals and approaches.

After Action Review Applicability. One viable area for inclusion of Intellimedia appears to be for after action reviews. Current after action review technologies assist human observers by recalling the training events through video, audio, and computer methods. In the realm of multiple media inputs for a structured presentation, an intelligent system will ease the indexing, browsing, retrieval and presentation of multimedia data (Maybury, 1994). Incorporation of an intelligent multimedia system would capitalize on capturing essential training points through accurate information storage and retrieval. A possible drawback is the difficulty which may be experienced when setting up all the equipment and projecting devices necessary to conduct a multimedia-based learning event.

Rapid Prototyping With Commercial Off-The-Shelf (COTS) Packages

The design team selected three COTS software packages for building the CATT TREDS initial prototype. These are Harvard Graphics for Windows (HGW, v 2.0), FoxPro for Windows (v. 2.5), and Microsoft EXCEL (v. 4.0). The programs feature object-oriented options for launching applications, storage and retrieval of data, incorporation of multi-media files, dynamic data exchange and object linking.

Harvard Graphics. The present technology demonstration model of CATT TREDS runs in the HGW screenshow mode. The primary reason for using the HGW software was its ease of use, shorter learning curve required and the familiarity with HGW by many military personnel. The main menu of the tool is seen in Figure 1. All underlying applications are launched from this main menu.

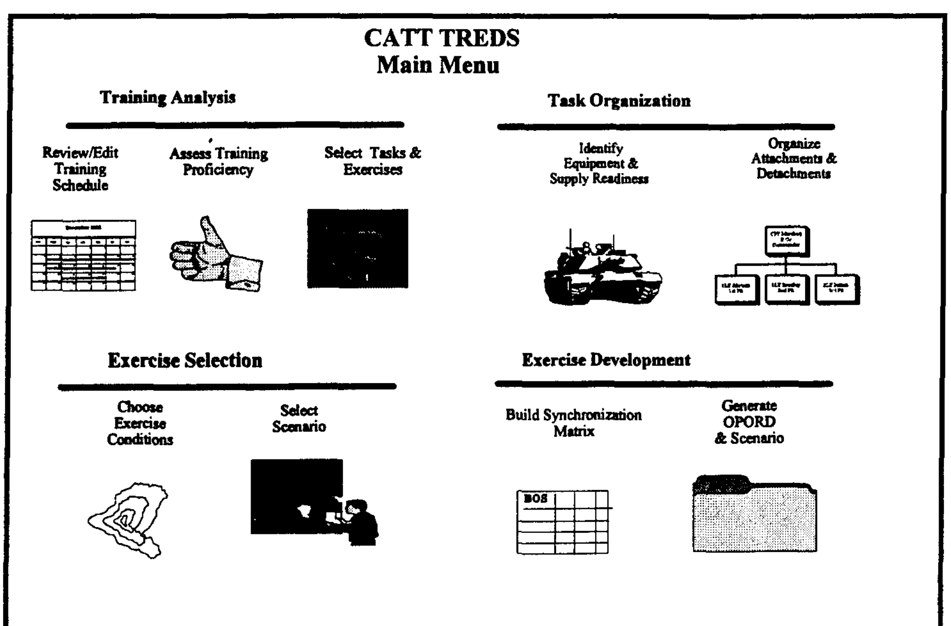


Figure 1. CATT TREDS Main Menu

Phase/Unit	TAA-Mike	Occupy Def. Sector	PL Gray	PL Yellow	Fwd of PL Silver	Consolidate
Est. Time	'0600Z	'0900Z	1200Z	1500Z	1700Z	1900Z
TM B/3-77 Mech	Move to TAA	Occupy Def Sector	Defend	1st PLT Moves back	CounterAttack, Support by Fire	
1st Plt B/3-77	"	",West fwd PL Silver	Defend in Zone	Occupy BP-33		
2d Plt B/3-77	"	", East fwd PL Silver	Defend in Sector, fwd of PL Silver		Control OBJ Melee	
1st Plt C/1-2 AR	"	", SE, South of PL Blue	Occupy BP 32		Attack along Axis Strike, into OBJ Melee	
Fire Support		Develop FS Plan	Submit FS Plan, Select FPF's	Main Effort 1st PLT	Support CounterAttack	
Mobility & Surv.	Clear Routes	Protect MSR, Emplace Obstacles	Emplace Obstacles			
Battle Command	Move to AA	Issue FRAGO's		Move 1st PLT	Order CounterAttack	Consolidate, Prepare for 2d Echelon
Logistics	Class 3, 5 & Maint. spt	Resupply in posn		Resupply 1st PLT		Consolidate
Intelligence	Recon Fwd Activity	Execute Recon Plan	Enemy at PL Gray	Enemy at PL Yellow	Enemy in OBJ Melee	

Figure 2. Synchronization Matrix

FoxPro. FoxPro application files are in runtime versions ready for retrieval and use by clicking on objects in the HGW screenshow slides. Presently two previously developed databases are used by CATT TREDS. They are CATT TASK and Equipment Characteristics Database, both developed for the Government by Resource Consultants, Inc. (RCI). Task, STX, trainability factors and proficiency rating data are incorporated from the former while equipment physical characteristics and capabilities are collected from the latter. The HGW and FoxPro merger demonstrates the easy and relatively seamless navigation that can be effected between FoxPro applications and other programs. FoxPro has the capability to import several database management systems programs for good versatility and adaptability in meeting user specifications.

Microsoft EXCEL. The synchronization matrix is the focal point for exercise implementation as it is the most widely used tactical planning tool. A standard EXCEL spreadsheet was chosen as the best tool in which to store the synchronization matrix data. Information from the spreadsheet is then Dynamic Data Exchanged (DDE) into the HGW shell. Subordinate units and predominant battlefield operating systems (BOS) are listed on the vertical axis, with events and estimated time lines along the horizontal axis. See Figure 2 above for an example format. Layout in the spreadsheet allowed the design team to capitalize on the database qualities of the matrix. The operations order (OPORD) will be stored as a series of database elements embedded in a text file unique to each scenario within the library. As the user makes changes in the synch matrix, the OPORD will be dynamically updated. The edit and update process has been designed to work in both directions. Changes made in specific fields in the OPORD are tied directly to fields in the matrix, as well as from the matrix to the OPORD.

ADVANTAGES AND DISADVANTAGES OF THE TECHNOLOGIES

Table 1 displays the initial evaluation findings of the design team for the above discussed technologies relative to use in an automated planning tool such as CATT TREDS. Criterion for the evaluation included cost of the technology, time required to develop the desired capability with the technology, anticipated acceptance by the target audience, known familiarity of the target audience with the technology, and ease of use in prototype development.

Table 1. Initial Technology Evaluation

Criterion / Technology	Costs	Time Required	Acceptance	Familiarity	Ease of Use
Expert Systems	Average/Low	Average	Average	Average	Average
Intellimedia	High	High	High	Low	Average/Low
Voice Recognition	Average	High	Average?	Average	Average/High
Decision Support Tools	Average	Average	High	Average	Average
Neural Networks	High	High	Average	Low	Low/Average
Rapid Prototyping	Low/Average	High	High	Average	Average

SYSTEM DESCRIPTION

CATT TREDS is a technology and concept demonstration development program sponsored by the Project Manager-Combined Arms Tactical Trainer (PM-CATT). The primary objective of the CATT TREDS initiative is to identify requirements, procedures and technologies to assist unit trainers in the development of training exercise plans for such CATT training systems as Close Combat Tactical Trainer (CCTT). The applicability of the tool for use with other simulation systems, for planning field exercises using actual equipment, and for other Service and joint force exercise planning has been shown to be a real possibility. CATT TREDS will aid unit trainers in many training situations to optimize their training opportunities given available resources.

Design Criteria

CATT TREDS utilizes the following design criteria:

- Windows for PC Operating Environment. The Windows environment allows dynamic data exchange and object linking between multiple software files.

- Commercial Off-The-Shelf Hardware and Applications Software. COTS hardware maintains pace with rapid technological change. COTS software reduces Government requirements to develop and maintain in-house computer programs.

- Object Oriented Design (OOD). OOD allows users to access a wealth of information in a single keystroke or click. Multiple applications can be embedded within a main overview application.

- Dynamic Data Exchange. DDE permits users and programmers to retrieve data transparently once links are established between a server and client application.

- User Centered Design. The human-computer interface makes the software easier to use during development, fielding, and maintenance of a system. and is one of its most noticeable features. The interface influences the ease of navigation between menus and submenus, command syntax, and editing capabilities (Babbitt, 1991). Using proven human-computer-interface techniques, an intuitively user friendly interface that is both flexible and easy to use can be provided to aid both the developer and the end user.

Capabilities

There are several categories of tasks that are essential to the training exercise planning process. They include training task analysis, task (resources) organization, exercise selection and exercise development. In addition, at the bottom of each screen are menu items to assist user navigation in CATT TREDS. The following capabilities are included by category in the CATT TREDS tool:

Training Analysis. The modules included here allow the unit commander to set up, review and edit the training schedule by day, month, and quarter, with rollup to battalion level or rolldown to platoon level. Training proficiency, in the form of Trained (T), Partially Trained (P) or Untrained (U) can be assessed and modified by training task and subtask. The Mission Essential Task List (METL) can be developed or modified. Trainability of the chosen training mode (i.e., specific simulator) relative to the particular task can be assessed, and training tasks can be chosen for the exercise with options provided to select or deselect as desired.

Task Organization. These modules allow the identification and assignment of organic unit resources. This includes the selection of equipment and details of equipment and supply readiness, and the organization of personnel attachments and detachments. Equipment capabilities for both friendly and enemy forces can be reviewed from an on-line library. The terrain database for the exercise can be selected. Factors identified here are used in the computer initialization of the simulation.

Exercise Selection. Capabilities included in these modules allow the unit commander to select the degree of difficulty for the exercise in the METT-T areas of enemy, troops, terrain, and time. Environmental conditions may also be selected. Degree of difficulty is measured as low, medium, and high resulting in over 63,000 possible combinations. Suitable scenarios complete with course of action overlays for battalion, company or platoon level may be selected from the available library as is without tailoring. A scenario from the library may also be modified, or if so desired, developed from scratch. All scenarios have the following items in common:

- Initialization data
- Established unit tactical graphics
- Enemy force organization for combat
- Graphical map of training area
- Friendly force organization for combat
- Any time compressed elements of the plan

Exercise Development. Using information developed in the various modules and employing the expert system and decision making technologies, a synchronization matrix such as that found in Figure 2 can be developed, modified as required. and generated. In addition, the OPORD and scenario course of action overlays can be generated. These products can be printed in hardcopy using an attached printer or by saving the appropriate files to diskette and using available print capabilities. By taking these CATT TREDS-generated tools to the training site, it is hoped that the unit commander may need only bring his own coffee cup in addition to be prepared for his planned exercise.

CONCLUSIONS FOR FUTURE TRAINING EXERCISE PLANNING

There are many techniques, tools and databases developed by industry and the military which can be applied to the training exercise planning process that allow unit planners to optimize exercise planning time. Incorporating them into a technology demonstration model like CATT TREDS has provided an opportunity to leverage these previously developed research applications into a user friendly training exercise planning tool. Probably the most exciting benefit to come from this development has been the possible uses that can be seen for other agencies. The concepts incorporated into CATT TREDS for Army use can be tailored for other Services, for joint exercises as well as those with multi-national forces, for exercises that entail not only virtual simulations, but constructive and live field exercises, and to the civilian community for disaster preparedness exercises, We have looked at only a few of the technologies that show promise for future use. Other multimedia techniques may meet more of the requirements required by training exercise planners in the future.

REFERENCES

Andrews, D. August 1993. IBM and Apple work to perfect voice input. *BYTE* p. 32.

Babbitt, B.A. (March 1991). Evaluation of a Rapid Prototyping Intelligent Tutoring System. Hawthorne, CA: Northrop Aircraft Division, NOR 91-43.

El-Najdawi, M.K. & Stylianou, A.C. (Dec 1993). Expert Support Systems: Integrating AI technologies. *Communication of the ACM*, Volume 36, No. 12. pp. 55-65.

Georgiopoulos, M. & Heilman, G. 1993. *Introduction to Neural Networks*.

Marchionini, G. & Crane, G. January 1994. Evaluating hypermedia and learning: Methods and results from the Perseus Project. *ACM Transactions on Information Systems*. Vol. 12, No. 1. pp. 5-34.

Maybury, M.T. 1994. Knowledge-based multimedia: The future of expert systems and multimedia. *Expert Systems with Applications*. Vol. 7, No. 3. pp. 387-396.

Morrison, J.D. September 1992. A 'neural network' model that supports realtime learning of temporal relationships in complex engineering domains. *Simulation 59:3.* pp. 152–163.

No Author. (April 20,1994). Grove pleads for multivendor OS. *Uninews* Vol VIII, No. 6. p. 3.

Ragusa, J.M. (1994). Models of multimedia, hypermedia, and 'intellimedia' integration with expert systems. *Expert Systems With Applications*, Vol 7, No. 3. pp. 407–426.

Thyfault, M. (May 1994), The power of voice. *Informationweek,* pp. 39–44.

University of Central Florida Institute for Simulation and Training. (March 1993). Distributed interactive simulation – operational concept. Orlando, FL: Institute for Simulation and Training.

AUTOMATED EXERCISE PREPARATION AND DISTRIBUTION FOR LARGE SCALE DIS EXERCISES

Barbara J. Pemberton, Douglas J. Classe,
Naval Air Warfare Center Training Systems Division
Orlando, Florida
Charles W. Bradley, Mike Wilson,
Hughes Surface Ship Division
Fullerton, CA

ABSTRACT

New automated approaches for preparation and electronic distribution of large scale Distributed Interactive Simulation (DIS) exercises is required to accommodate the increasing number of DIS exercises and geographically dispersed exercise participants.

This paper describes two prototype tools -- 1) automated DIS exercise preparation tool, and 2) an automated electronic distribution tool. The preparation tool uses an expert system to reduce the time to prepare large scale DIS exercises from weeks/months to minutes/days. The electronic distribution tool demonstrates a first implementation of the DIS "Set Data" protocol data unit (PDU) for electronic exercise initialization.

Three viewpoints of the automated tools are combined in this paper: 1) government -- requirement statement, and DIS implementation, 2) contractor -- systems analysis and expert system implementation, and 3) military -- ease of use, validation.

Future direction and joint applications of the automated DIS tools are also presented.

ABOUT THE AUTHORS

Barbara J. Pemberton is a Computer Scientist for the Department of the Navy at the Naval Air Warfare Center Training Systems Division. Mrs. Pemberton is currently the Principal Investigator for research of automatic exercise preparation/distribution and control using expert systems, Mrs. Pemberton holds a Master of Science in Management from Rollins College and a Bachelor of Science in Mathematics from the University of Tennessee. Her previous experience includes training system software acquisition and development at General Electric and Martin Marietta.

Douglas J. Classe is an Electronic Engineer for the Department of the Navy at the Naval Air Warfare Center Training Systems Division. He is currently responsible for system design and implementation of the automated DIS exercise preparation and distribution tool. He holds a Bachelor of Science Degree in Electrical Engineering from the University of Central Florida. His previous experience includes Department of Defense software acquisition for the Navy, and high frequency amplifier design using GaAs technology at Martin Marietta.

Charles W. Bradley is a Senior Systems Engineer with Hughes, Surface Ship Division. Mr. Bradley is responsible for expert system research. Prior to his employment with Hughes, Mr. Bradley worked for nine years developing, evaluating, and documenting tactics for new weapon systems as they were introduced into the fleet. Prior to his civilian employment, Mr. Bradley served for nine years as a commissioned officer in the U.S. Navy and was a qualified Surface Warfare Officer. Mr. Bradley has a Masters of Science degree in Operations Research from the U.S. Naval Postgraduate School, and a Bachelor of Arts degree from the University of Alabama.

Michael F. Wilson is a software engineer with Hughes, Surface Ship Division. Mr. Wilson has over 20 years of software experience. His most recent work includes the use of expert systems and graphical user interfaces to design and implement an automatic tool for exercise preparation for large scale DIS exercises. Mr. Wilson has a Bachelor of Science degree in Electrical Engineering from the University of Washington.

AUTOMATED EXERCISE PREPARATION AND DISTRIBUTION FOR LARGE SCALE DIS EXERCISES

Barbara J. Pemberton, Douglas J. Classe,
Naval Air Warfare Center Training Systems Division
Orlando, Florida
Charles W. Bradley, Mike Wilson,
Hughes Surface Ship Division
Fullerton, CA

INTRODUCTION

Today's military strategy has changed from a focus on a global threat to a focus on multiple regional conflicts [1]. Regional conflicts consist of confined and congested water and air space occupied by friends, adversaries, and neutrals. Rapid *preparation* and *distribution* of training exercises in a common Distributed Interactive Simulation (DIS) networked environment (Figure 1) is required for quick coordinated action by all forces (e.g., joint and/or coalition). The forces participating in a DIS exercise will be using many different types of systems with different exercise specification and initialization needs -- man-in-the-loop training simulators, embedded training systems, wargaming simulators and live ranges.

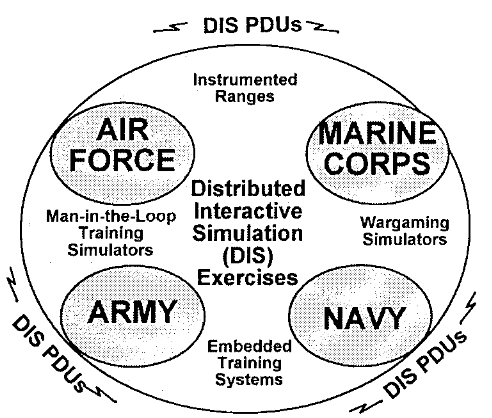


Figure 1. A DIS Training Environment

Today, creating training exercises is time consuming and labor intensive. The exercises are difficult to modify or vary in response to dynamically changing training requirements. In some cases, an exercise with 2000 objects (e.g., platforms, personnel) requires from weeks to months to prepare. Large scale joint/coalition exercises will require 10,000 to 100,000 or more objects. This implies a potential corresponding five to fifty-fold increase in the time to prepare a large scale exercise. Also, in current DIS demonstrations, the exercise initial conditions are manually entered by each participant into their system. A manual approach can easily lead to errors in platform placements and is a slow process.

Automated tools are required to -- 1) reduce exercise preparation time from *months/weeks* to *days/hours*, and also 2) electronically transfer in *minutes* 100,000 or more exercise objects to all DIS participants

OBJECTIVE

The objective of this paper is to describe two prototype tools: -- 1) Automated Exercise Preparation, Evaluation, and Preview (APEP) Tool-- to reduce exercise preparation time, and 2) Automated Exercise Distribution and Display (ADD) Tool -- to electronically distribute large scale DIS exercises to exercise participants (see Figure 2).

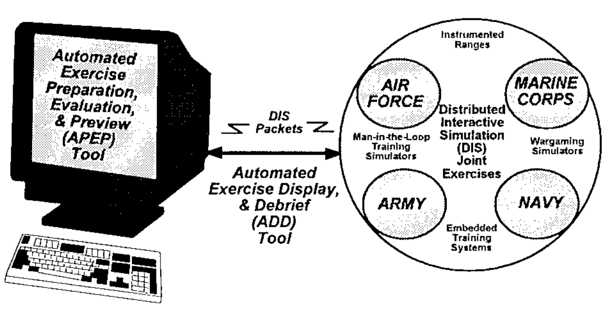


Figure 2. TWO TACTICS TOOLS -- Automated Exercise Preparation, Evaluation & Preview (APEP) Tool, and Automated Exercise Display & Debrief (ADD) Tool

The APEP Tool utilizes expert system technology and the ADD Tool utilizes simulation management DIS protocol data units (PDUs). A description of the two prototype tools, an evaluation of the APEP tool, and future expansion of the tools is presented in the following paragraphs.

AUTOMATED EXERCISE PREPARATION, EVALUATION, & PREVIEW (APEP) TOOL

The Automated Exercise Preparation, Evaluation, and Preview (APEP) Tool (Figure 3) overall concept consists of three capabilities: automated exercise force laydown based upon a specific training objective, 2) automated platform scripting using computer generated forces, and 3) automated association of training objectives with performance measurement criteria.

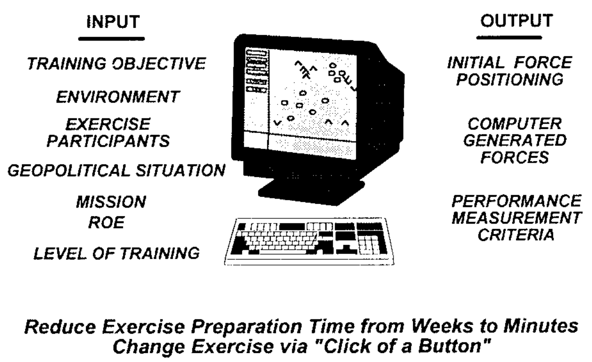


Figure 3. Automated Exercise Preparation, Evaluation, and Preview (APEP) Tool Overview

APEP Tool Prototype -- Training Exercise Force Laydown (TEFL)

The first APEP Tool capability prototyped is called Training Exercise Force Laydown (TEFL). TEFL is a proof-of-concept effort to demonstrate that one Training Supervisor can develop training exercises that:

(1) Represent non-trivial tactical situations, and comprise a relatively large number of both friendly and hostile units of all platform categories (e.g.,. surface, subsurface, and air).

(2) Are random enough in friendly and hostile unit placement to avoid the problem of trainees being able to unreasonably predict what will happen during training;

(3) Employ force laydowns which are tactically sound, both from a friendly force and a hostile force point-of-view;

(4) Support specific training objectives selected by the Training Supervisor;

Furthermore, this single Training Supervisor can obtain these exercise force laydowns with the TEFL within *minutes,* instead of the *weeks/months* required with the current manual processes.

The basis of the TEFL concept is to employ Artificial Intelligence techniques - specifically expert systems - to provide automation for training exercise force laydown. The Training Supervisor specifies only the "kind" of training he wants to conduct in high level, abstract, descriptive terms, and then TEFL's embedded expert system automatically infers the details of both the friendly BLUE and hostile RED Force laydown.

How a Training Supervisor Uses TEFL

First, the TEFL Training Supervisor selects a Training Objective from a menu of eight. These include:
(1) Strike Ashore
(2) Initial Approach from Seaward
(3) Long Range Battle Group Anti-Air Warfare
(4) Short Range Battle Group Anti-Air Warfare
(5) Coordinated Battle Group Anti-Air Warfare
(6) Submarine vs Ship Anti-Submarine Warfare
(7) Submarine vs Submarine Anti-Submarine Warfare
(8) Coordinated Battle Group Anti-Submarine Warfare

Next, the Training Supervisor selects BLUE and RED ships, submarines, and aircraft (by specific hull, pendant, or side number) to be included in the training exercise. TEFL employs a Technical Data Base indicating the specific sensors, weapons, and combat support equipment for each platform.

Finally, options specifying tactical constraints and limitations are specified by the Training Supervisor (e.g. initially Hot or Cold state-of-war; use of nuclear weapons possible; heavy jamming environment).

When all of the exercise descriptors have been input, the Training Supervisor initiates the TEFL expert system by selecting the menu button LAYDOWN. TEFL then automatically determines a laydown position - an exercise start position - for each of the BLUE and RED units in the specified exercise forces. Initial course, speed, altitude (for aircraft), depth (for submarines), sensor employment, and weapons status are also determined. Units are automatically plotted on the TEFL PPI display. Tabular, alpha-numeric displays of unit position and kinematics status can also be displayed. Both displays can be printed.

TEFL -- "Click of a Button" Exercise Variability

The Training Supervisor may save the current exercise and re-load at a later time. If the TEFL menu laydown button is "clicked," a new exercise force position will be computed that will still accomplish the original training objective. The TEFL system currently can vary any one or more of 17 parameters (e.g., battle group speed, range, RTF bearing) and maintain the original training objective. This feature of "click of a button" variability makes preparing initial force positioning for a training exercise occur in minutes.

Knowledge for the RED Force Laydown expert system was obtained from "Soviet Naval Tactics," Dr. Milan Vego; Naval Institute Press; 1992 [2]. Dr. Vego has 12 years commissioned service in the Yugoslav Navy, and during that period, worked closely with the Soviet Navy. He defected to the United States from Yugoslavia and has since worked closely with the U.S. defense establishment. Dr. Vego's book deals with Soviet Naval Tactics when the ex-Soviet Union was the principal threat to the United States. He maintains that countries with exported Soviet or Russian ships, aircraft, submarines, sensors and weapon systems are very likely to employ tactics closely patterned on those outlined in his book.

TEFL Expert System Implementation

TEFL is implemented using a commercial off-the-shelf (COTS) expert system shell [3], and graphical user interface (GUI) [4] The TEFL system components are shown in Figure 4. A TEFL knowledge base is defined as a file that contains both rules and objects. There are three main TEFL knowledge bases -- 1) Top Level Force Laydown, 2) BLUE Force Laydown, and 3) RED Force Laydown.

Top Level TEFL Control Expert System Knowledge Base

The first expert system controls top-level TEFL operation based on Training Supervisor inputs specifying the BLUE and RED Forces; the exercise training objective; and any desired operational limitations and constraints. It controls the overall geometry of the placement of the RED Target on the playing area, the direction from which BLUE approaches RED; and the distance of the BLUE Force Center (BLUE Zulu Zulu) from the RED Target.

BLUE Force Knowledge Base

Once the RED Target, the RED Airfield and the BLUE Force Center has been placed. The individual BLUE ships submarines and aircraft can be placed about BLUE Force Center.

RED Force Knowledge Base

Once the BLUE forces have been laid down, then the RED forces are laid down. Depending on the exercise objective, some of the RED units are placed with respect to the BLUE forces (e.g. the RED surface and subsurface tattletales are placed around the BLUE CV).

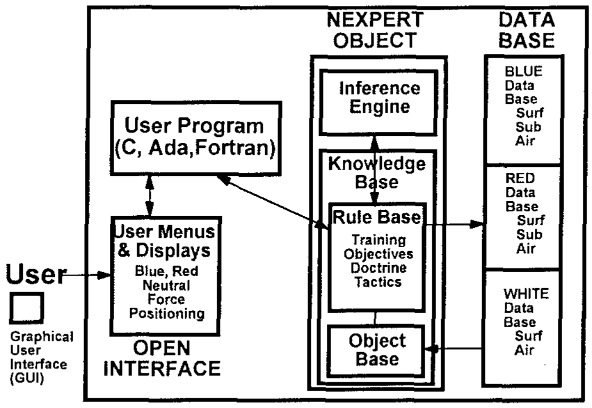


Figure 4. APEP Tool Software Components

Directions for Future TEFL Development

The current experimental concept TEFL successfully demonstrated that a single Training Supervisor can quickly and easily generate RED and BLUE Force laydowns for relatively large, tactically non-trivial exercises which provided variability in training and are tailored to specific training objectives. Nevertheless, there is still additional TEFL development work needed to provide full exercise generation and execution support for shipboard Training Supervisors. TEFL features that need to be added include:

(1) Expanding the Expert System to address unit laydown for additional warfare areas (other than the current CV BG Strike Ashore area) to include: (a) amphibious operations; (b) mine laying, mine hunting, and mine sweeping; (c) cruise missile (e.g. TLAM) strike ashore; (d) war at sea exercises (including long and medium range over-the-horizon targeting (OTH-T) and surface ship anti-ship cruise missile (ASCM) attack); (e) convoy escort; and (f) combined warfare including elements of all warfare areas.

(2) Increasing the number of RED and BLUE platform classes so common platforms from each service can be included in Joint Force training exercises.

(3) Incorporating map data bases so that force lay downs could be generated with respect to geographic location of forces and the proximity of cities, naval bases, air fields, industrial facilities, roads, railroads, political boundaries, and geographic features.

(4) Integrating TEFL with standard naval tactical and intelligence data bases (e.g. Naval Warfare Tactical Database (NWTDB)).

(5) Add neutral WHITE ships, aircraft and even submarines that could serve as background platforms in the environment to the BLUE and RED combatants.

(6) Allow the operator to specify time-of-day and weather conditions (e.g. wind speed and direction, sea state, atmospheric temperature profile with altitude, bathythermograph data, cloud cover).

(7) Provide a Computer Generated Forces (CGFs) capability for automated platform scripting. This means the Training Supervisor would not have to manually script each platform. The Training Supervisor could then preview the likely training exercise outcomes based upon CGF movements. Additionally, performance measurement criteria to be gathered, stored, and computed in real-time for instructor display and debrief can be identified. With the availability of CGFs, any platform not modeled by a DIS exercise participant could be modeled by the CGFs of TEFL.

System Evaluation

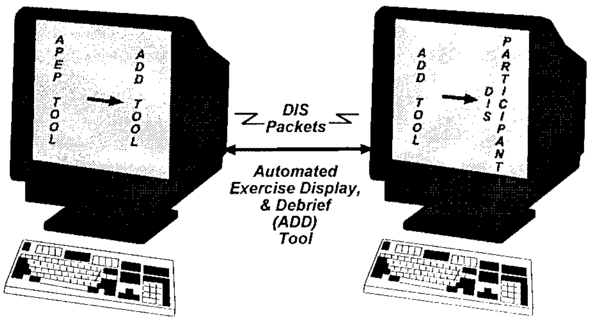
The TEFL automated exercise force laydown prototype tool is being evaluated by both naval active duty and reserve personnel (see ADVISORS), academia, and NAWCTSD research and engineering personnel.

Navy military personnel have independently validated each of the TEFL training objectives and subsequent force laydowns with an optimum force composition for both BLUE and RED Forces. The need to add the features listed under Directions for future TEFL Development were independently derived by the Navy personnel and TEFL developers. Additional features desired by the Navy personnel include: 1) allow the selection of aircraft squadrons instead of the selection of individual aircraft, 2) the ability to display sensor, and weapons inventory by platform.

For quicker verification of the TEFL expert system rule base, an automated tool is needed to check for: 1) redundant rules, 2) conflicting rules, 3) subsumed rules, 4) circular rules, 5) unnecessary IF conditions, 6) Dead-end rules, 7) missing rules, and 8) unreachable rules [5].

AUTOMATED EXERCISE DISTRIBUTION & DISPLAY (ADD) Tool

The second tool, Automated Exercise Distribution and Display (ADD) tool, electronically distributes the output of the APEP tool to the DIS training exercise participants using DIS protocol data units (PDUs). This tool is used after the APEP tool has created a battle force laydown for a specific training objective (see figure 5). A "handshaking" paradigm between two ADD tool processes using DIS simulation management PDUs was designed to implement a quick method of electronically transmitting the initial training exercise data to each DIS exercise participant.



Electronically Distribute Large Scale DIS Exercises

Figure 5. ADD Tool

Initialization Data Requirements

There is basic initialization information which all simulators require at start up. Initial position, and an initial velocity vector are examples of initialization data required by almost all simulators. However, there is also data which is simulator dependent (e.g., terrain data, environmental conditions, mission, and rules of engagement). Implemented in the ADD tool prototype is the transfer of platform type, platform mission, hull number, initial position, and initial velocity.

DIS PDU Selection

Discussions were held between NAWCTSD and the DIS Simulation Management Subgroup on which PDUs are best suited for electronic training exercise distribution. The conclusion was that a tool for electronically transferring training exercise preparation data had not yet been implemented using Simulation Management DIS PDUs. The guidance received from the DIS Simulation Management Subgroup was to implement the ADD tool using any PDU(s) which seem appropriate, however the message PDU should only be used for documentation. It was later decided by NAWCTSD that the *action request, action response, and set data* PDUs would be used in the ADD tool prototype.

System Design Issues

Several System level design issues surfaced during the design and implementation of the ADD Tool prototype. Two of these are:

1) *Distribution of a single exercise to a single exercise participant* -- The initial ADD Tool effort was targeted to downloading a single APEP training exercise to a single DIS training exercise participant. This effort enabled a quick implementation and test of the handshaking algorithm. Issues dealing with multiple exercise participants were left to be solved later.

2) *Distribution of a single exercise to multiple exercise participants* -- Downloading a single training exercise to several DIS exercise participants or several different simulation host computers has been investigated.

The first problem to overcome is how to assign simulation entities to simulators. There are two parts to this problem: a) how to assign each platform to the most appropriate simulator, b) how to handle the discrepancy of a simulator requesting to participate in a training exercise when no appropriate platforms are available in the current APEP training exercise.

The second problem is to resolve the inconsistencies in the initialization data requirements of heterogeneous DIS simulators participating in a common DIS training exercise.

High Level Implementation

A handshaking paradigm was designed using the DIS action request and action response PDUs. The action request PDU is used to initiate the download process. Action response PDUs are used to respond to the action request PDU, as described in the DIS 2.0.3 standard [6]. Embedded in the handshaking scheme is the transmission of DIS set data PDUs. The set data PDUs carry the exercise initialization data described earlier.

Detailed Implementation

The ADD Tool consists of two major functions -- ADD sender and ADD receiver. The ADD sender software physically resides on the APEP Tool hardware platform. The ADD receiver software physically resides with each DIS exercise participant's simulator (see Figure 6).

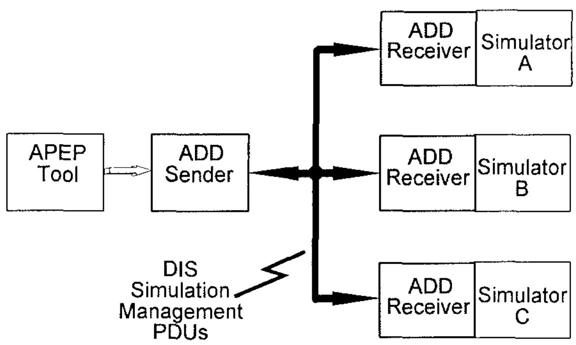


Figure 6. ADD Tool Architecture

Upon invocation of the ADD tool, an ASCII output file from APEP is read and parsed into the ADD tool's data structures. Once all the training exercise preparation data is loaded into memory, the start of the DIS simulation management handshaking paradigm begins. To initiate the handshaking algorithm, an action request PDU is sent from the ADD sender to the ADD receiver. The action id field in the action request PDU is filled with an action id enumeration equal to "RECEIVE_SCENARIO." This enumeration is an extension to the DIS 2.0.3 standard and is required to complete a DIS training exercise download.

The ADD receiver responds by sending an action response PDU back to the ADD sender. Within the action response PDU, the request status enumeration field is set to "PENDING." This enumeration is in accordance with the DIS 2.0.3 standard. The sender then knows the receiver is ready to accept the exercise preparation data.

Three set data PDUs are then sent by the ADD sender to the ADD receiver. The first set data PDU contains high level BLUE and RED Force information, the second set data PDU transmits detailed BLUE platform data, and the third contains detailed RED platform data (see table 1). Lastly, an action response PDU with the request status enumeration field set to "COMPLETE" is sent from the ADD receiver back to the ADD sender. The handshaking algorithm now is complete and the ADD sender is ready to download the training exercise data to another simulator. A summary of the ADD Tool operation is shown in Figure 7.

First Set Data PDU		
	BLUE Force Center Location	x, y, z
	BLUE Force Velocity Vector	speed, heading
	RED Force Center Location	x, y, z
	RED Force Velocity Vector	speed, heading
Second Set Data PDU		
	For Each BLUE Platform	platform type
		mission
		hull number
		location x
		location y
Third Set Data PDU		
	For Each RED Platform	platform type
		mission
		hull number
		location x
		location y

Table 1. Contents of Set Data PDUs

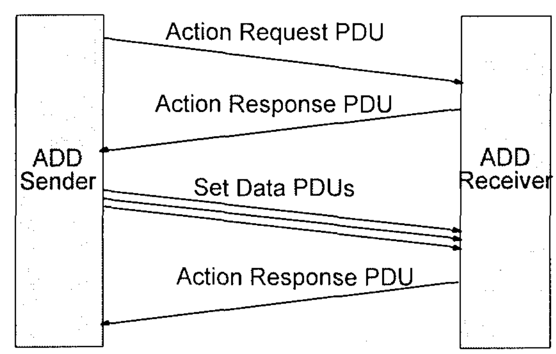


Figure 7. ADD Tool Exercise Download

Add Tool Statistics

The ADD tool prototype contains over 2,200 lines of C source code. The prototype was designed with a functional decomposition approach using a UNIX C compiler/linker and the vi editor. Approximately 700 lines of code were opportunistically reused from another NAWCTSD research project. The ADD tool prototype currently operates on a SUN SPARC under the SUNOS 4.1.2 operating system.

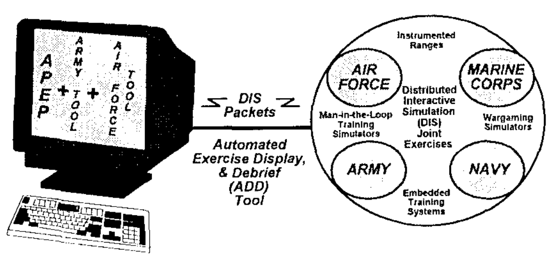
ADD Tool Future

Projected future enhancements to the ADD prototype include: 1) exercise download to a DIS simulation manager ; 2) support for multiple levels of DIS simulation management; 3) integration with the Naval Warfare Assessment Division's Warfare Assessment Model (WAM) modified for real-time display and debrief.

FUTURE APPLICATIONS -- APEP, ADD TOOLS

The APEP Tool is currently being used for evaluation by the NAWCTSD research and engineering department and military advisors. The tool is expected to be demonstrated at the 16th I/ITSEC DIS demo. Additionally, Integration with the Battle Force Tactical Trainer (BFTT) demonstrations is in progress.

To support joint exercises, the APEP tool can be integrated with other services' exercise preparation tools (Figure 8). An initial investigation of the Army's RASPUTIN exercise preparation tool indicates that the two tools could be joined to accomplish joint exercise preparation.



Reduce Exercise Preparation Time from Weeks to Minutes

Electronically Distribute Large Scale DIS Exercises

Figure 8. Joint DIS Exercise Distribution and Display

Although the ADD tool currently operates with the APEP prototype tool, the concepts and software can be applied to future joint/coalition systems.

Display and debrief of joint DIS exercises will initially involve each service's normal display techniques, however, a unified joint display and debrief tool will eventually be necessary.

REFERENCES

[1] "From the Sea - A New Direction for the Naval Service," a Navy and Marine corps White Paper

[2] "Soviet Naval Tactics, "Milan Vego, Naval Institute Press, 1992

[3] NEXPERT OBJECT ™ -- Expert System,. Neuron Data, Palo Alto, CA

[4] OPEN INTERFACE ™ -- Graphical User Interface (GUI),. Neuron Data, Palo Alto, CA

[5] "The Engineering of Knowledge-based Systems", Avelino J. Gonzalez and Douglas D. Dankel

[6] -- "Proposed IEEE Standard Draft Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications, " Version 2.0.3, STRICOM, DMSO

ADVISORS

Cdr. Neil Tollefsrud USNR-R graduated from the U.S. Naval Academy in 1975 and has a Bachelor of Science in Systems Engineering and Masters of Engineering Administration. He served onboard the USS Biddle CG-34 and at the Fleet Combat Training Center Atlantic. He currently works for Martin Marietta as a staff systems engineer in Orlando, FL.

Cdr. Ronald W. Wetmore USNR-R graduated from the United States Naval Academy in 1976 with a BSEE. He served on active duty for 6 years and qualified in Submarines. As a reserve officer he has supported NAVSEASYSCOM and served as exercise support for both U.S. and NATO exercises. Cdr. Wetmore hold a Master of Science in Nuclear Engineering from MIT and a Masters of Business Administration from Florida Tech. He currently works for Martin Marietta as Manager of External Tank Operations for the Space Shuttle at Kennedy Space Center.

Lt. Rodney A. McAvoy USNR-R graduated from the University of Mississippi with a bachelors degree in Marketing. He served as a surface warfare officer onboard the USS Forrestal and USS Dale. At the Naval Air Warfare Center Training Systems Division, he was an assistant project manager for a fleet battle group training system.

He currently works as a Regional Manager for Waterpoint in Fort Worth, Texas.

Lt. Jeff Weinrich USN is a 1984 graduate of Jacksonville University, with a BA in History. He has served onboard the USS Patterson (FF-1061), USS Spiegel Grove (LSD-32), at Assault Craft Unit Two, and Navy Support Facility Diego Garcia. His current duty is at the Naval Air Warfare Center Training Systems Division as the assistant project manager for Surface Propulsion Training Devices.

Applying Artificial Neural Networks to Generate Radar Simulation Data Bases

Harry H. Heaton III
Science Applications International Corporation
Dayton, Ohio

ABSTRACT

Modern combat aircraft sensor systems such as synthetic aperture radar (SAR) produce highly detailed, information rich displays. The simulation of such displays for training has demanded ever increasing computational resources as well as data sources more detailed than normally available digital feature analysis data (DFAD). By focusing on the correct reproduction of the content of a radar display rather than on a detailed model of radar physics, a novel Digital Radar Land Mass Simulator (DRLMS) for training is briefly described. A prototype of the system reproduces realistic real-beam, Doppler beam sharpened (DBS), and SAR ground maps from readily available data sources.

This radar simulation technique depends upon highly detailed, modified phototexture databases which contain both dimensional and effective radar cross-section information for broad area clutter and specific radar targets. This paper discusses the application of artificial neural networks in generating such databases from readily available data sources including Project 2851 and commercial satellite data. The issues, differences and solution approaches necessary to generate databases from such disparate sources as overhead imagery, DFAD feature data and existing simulator visual system databases are examined.

The techniques discussed have broad applications to the low-cost simulation of imaging sensor displays including millimeter microwave (MMW) and forward looking infrared (FLIR). The approach also drastically reduces the computational needs for a DRLMS system. The prototype, capable of generating SAR maps, was hosted on a single Motorola 68040 processor in a Macintosh personal computer. A simulation of the APG-68 radar, including real beam, expanded and DBS modes, is targeted to run in real time on a single MIPS R-4400 microprocessor.

ABOUT THE AUTHOR

Mr. Heaton has been an employee of Science Applications International Corporation (SAIC) for the past ten years, where he is the manager of the Microprocessor Processor Applications Branch at SAIC's Aeronautical Systems Operation in Dayton, Ohio. Mr. Heaton has been involved in the development of several simulation systems, including the B-1B Engineering Research Simulator, the B-1B High Fidelity Defensive Crewstation Simulator, and the Ground Support System for the F-16 On-Board Electronic Warfare Simulator. He is currently the program manager for an effort to field a low-cost ground map radar simulator for the APG-68 radar used by the F-16 aircraft.

INTRODUCTION

The Digital Radar Land Mass Simulator (DRLMS) discussed below departs from the traditional approach used for most DRLMS. Rather than relying on a rigorous simulation of the electromagnetic properties of the radar, the approach below uses photographic imagery pre-processed using a neural network to yield a database of pre-assigned radar reflectivities, which are then manipulated in real-time to yield a simulated high resolution Synthetic Aperture Radar (SAR) patch. For the engineer familiar with the physics of radar systems, this approach may seem too simplistic a process to yield credible results. Yet the images produced, while not to be confused with a target signature prediction, are convincing for man-in-the-loop training and laboratory applications.

One factor that allows the simplified DRLMS approach is the nature of image formation in high-resolution radar, where the effects of aspect are reduced in comparison to low resolution systems. This reduction in aspect sensitivity allows effective simulated imagery to be generated from simplified processes.

HIGH RESOLUTION RADAR TARGETS

For ground mapping radar, nearly all point targets extend across several wavelengths, and are said to occupy the optical region, where the ray tracing methods of geometric optics can be applied to estimate the radar cross section (RCS). A radar target is composed of one or more scatterers, depending upon the nature of the target and the radar. For the purpose of RCS estimation, these individual scatterers are described using various geometric shapes that allow ray tracing. Several of the fundamental shapes and their effective RCS (σ) in the optical region are defined below:

Sphere of radius a $\sigma = \pi a^2$

Normal to flat plate of areaA $\sigma = \dfrac{4\pi A^2}{\lambda^2}$

Normal to triangular corner

reflector of edge length a $\sigma = \dfrac{4\pi a^4}{3\lambda^2}$

The spherical scatterer is considered to be isotropic, and thus presents the same radar cross section regardless of the incidence angle of illumination. For flat plate reflectors and complex reflectors made up of flat plates, such as a corner reflector, the return strength varies depending upon the incident angle. The RCS of a flat plate exhibits a sensitivity to aspect (directivity) that is proportional to the size of the reflector. Table 1 depicts the mainlobe null-to-null angle for flat plate reflectors of increasing size. Note that for small reflectors, the useful angle is quite large.

Table 1:
Flat Plate Directivity vs/ Size (X-Band)

Flat Plate Size (m)	Flat Plate Size (λ)	Null-to-Null (degrees)
0.1	3	≈ 38*
.3	10	≈ 12
3	100	≈ 1
30	1000	≈ 0.1

* value shown does not consider resonant effects applicable to targets of less than 10λ.

Complex reflectors such as trihedral corner reflectors are even less sensitive to aspect. Large corner reflectors (such as found in many man-made objects) presenting strong returns over incident angles of roughly 45¡.

For a high-resolution X-band radar with a resolution cell size of 1 meter, each cell can only contain a few optical region scatterers (the cell is only $\approx 30\lambda$ on a side). Since the most probable scatterer size in a cell is small in terms of λ, the mainlobe size is correspondingly large, resulting in a return that is relatively constant over large aspect angles. This is the case found in practice with high resolution radar, where scattering sources are found to remain fixed in target location over aspect variations up to about 30¡ [1]. The small size of the cell in λ also increases the probability that a single scatterer dominates the return.

In contrast, a low resolution radar, where each resolution cell covers a very large area relative to λ, contains multiple scatterers in the optical region. First, it is now possible for very large scattering facets to exist within the range cell, resulting in strong, but highly directive returns.

Also, the return from multiple reflectors within the range cell is the phasor sum of the individual returns. This sum varies depending upon the geometry between the radar antenna and the relative phase between the individual scatterers. This phasor sum can vary greatly as the aircraft moves relative to the target, resulting in fluctuating return intensities that are highly aspect sensitive. This case is illustrated in Figure 1, where an identical set of corner reflectors is illuminated by both a high resolution and a low resolution radar. While the low resolution radar return is dependent upon the phasor sum, the high resolution radar sees relatively constant returns from fixed locations over large aspect angles.

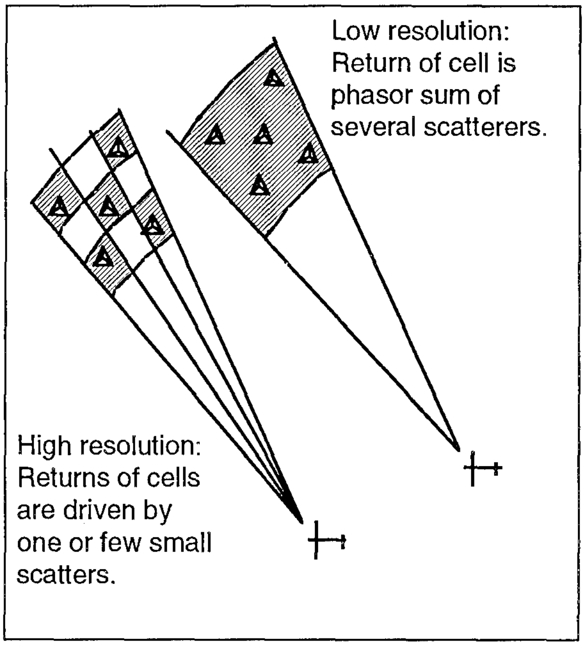


Figure 1: Comparison of return from high and low resolution radar from discrete corner reflectors.

The astute reader may wonder by this point, "What on Earth has this got to do with neural networks and DRLMS data bases?" The point of the preceding discussion is to identify the underlying reasons for the fundamental differences in the displayed returns from real-world targets produced by both high and low resolution radar systems. By capitalizing on the fact that the spatial arrangement of scatterers that are relatively insensitive to aspect (compared to low resolution radar) it is possible to use readily available photographic image sources to construct data bases for effective high resolution radar simulation. The photographic nature of the data inherently provides the spatial arrangement of features.

Neural networks can be applied to identify material classes from the spectral signature of readily available multi-spectral imagery. This can in turn, now be assigned a suitable value to represent radar reflectivity. In our simplified DRLMS process, we have applied this to imagery on a pixel-by-pixel basis, allowing a detailed correlation between a simulated SAR patch and the original photographic data source.[2] Below, the use of this approach is described and contrasted to traditional DRLMS processing.

TRADITIONAL DRLMS APPROACH

Conventional DLRMS systems have relied upon the availability of digital representations of terrain and cultural features. The terrain is maintained as DTED terrain posts and cultural data is derived from DFAD text files. A propagation model determines the two-way losses for each pulse as it travels to and from terrain and features, calculating the phasor sum for the signal at the receiving antenna. The energy reflected from the terrain is determined by ray-tracing each pulse, taking into accounting for such attributes as slope, material type, and terrain texture and correcting for receiver attributes such as sensitivity time control (STC) and gain. Because of the complexity of these calculations, DRLMS systems have been synonymous with complex, special purpose hardware in order to maintain real-world display update rates. Extending this approach to high resolution radar such as SAR continues to challenge the state-of-art in computing.

A key limitation for high resolution DRLMS is the content of the underlying database[3]. Advanced signal processing technologies allow radar such as Doppler beam sharpened (DBS) and synthetic aperture radar (SAR), to achieve resolutions on the order of a 5 meters or less for tactical systems. Standard database products have not kept pace. Level 2 DTED for example, provides terrain at 30 meter resolution, and Level 2 DFAD allows cultural resolution to two meters. While suitable for some high resolution radar simulations, the Level 2 DFAD product has limited geographic availability. Military training simulators frequently have a world-wide mission requirement.

To fill the gap between the expected content of the simulated sensor display and the content of the typical database, generic patterns are often applied. The major drawback of this approach is the loss of geo-specific content in the data base. Unless modified by specific cultural features, a SAR patch of one urban area will look much the same as the next, and may bear little resemblance to the real-world.

PHOTOTEXTURE BASED DRLMS

Phototexture is the term applied here to multi-spectral aerial images. (The term "phototexture" comes from visual simulation where it describes photographic data that is applied to polygon surfaces to enhance realism.) Such imagery is readily available, and is included as part of the Project 2851 SIF format. This same phototexture data can be applied as the starting point in generating representative high resolution radar imagery. Typically, these images are obtained from commercial satellite or aerial photography.

Prototype system description

SAIC has developed a prototype DRLMS system that demonstrates the utility of this approach to high-resolution radar simulation. The prototype radar simulation system is hosted on a commercial Macintosh Quadra computer (68040 @ 25 MHz). No special purpose hardware is required, allowing for re-hosting on a variety of systems. Pre-processing of the raw phototexture data followed by selective real-time image processing is the core of the system. The overall process for generating a SAR image is depicted in block diagram form in Figure 2. With additional processing to simulate fluctuating target returns and other effects, the capability has been extended to DBS and real-beam displays as well.

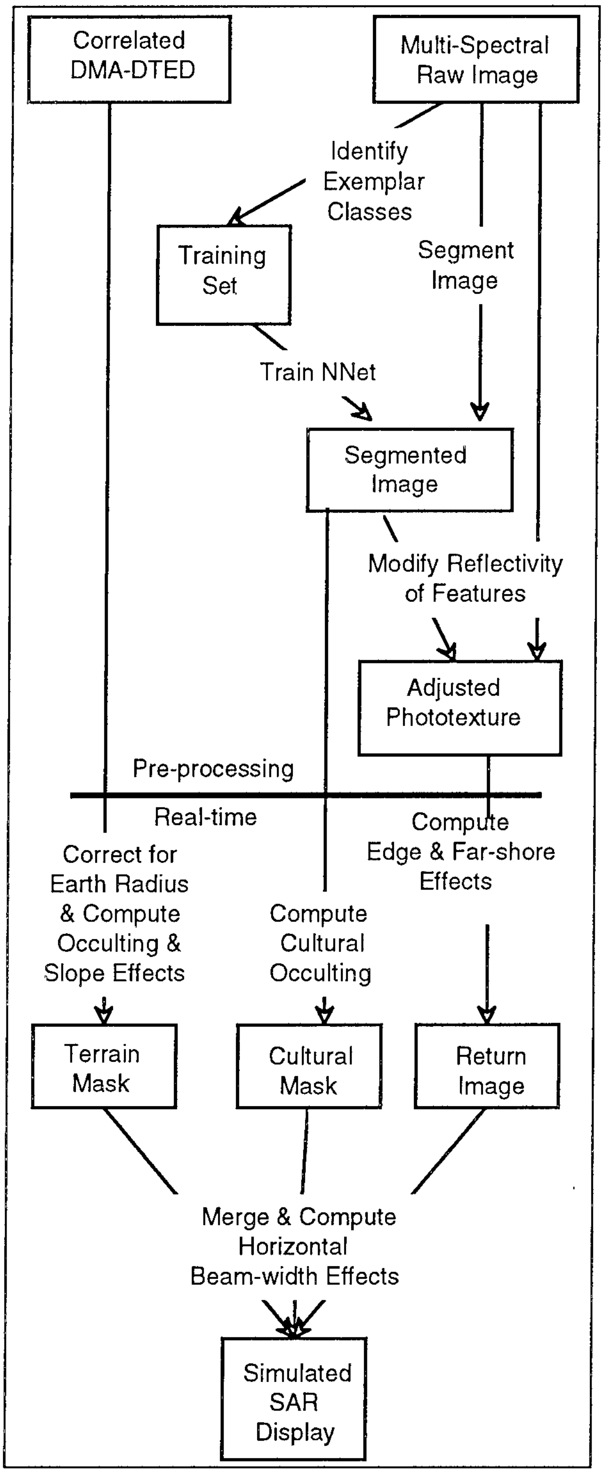


Figure 2. Simplified block diagram of phototexture-based radar simulation process. Processes depicted above the heavy line are pre-processing. Operations below are performed in real-time.

Database preparation overview

In comparison to Level 2 DFAD, high resolution photographic imagery is available on virtually a world-wide basis. Remote sensing satellites such as SPOT routinely generate multi-spectral imagery with a resolution of 20 meters, and panchromatic imagery to 10 meters. Imagery sold commercially from Russian satellites with the MK-4 camera provide multi-spectral six meter data. Recent easing of U.S. restrictions on the resolution of commercial satellite imagery foreshadows a blossoming market for sub-10 meter imagery. In addition to satellite imagery, false color and multi-spectral imagery from aircraft offers affordable access to sub-meter photographic data sources.

The DRLMS simulation process begins with the segmentation of a multi-spectral photographic image into feature classes. Segmentation is the process of breaking a complex data set, such as a multi-spectral image, into distinct sub-classes. This is accomplished by training a feed-forward, back-propagation neural network to classify individual pixels based upon their spectral content in the available bands[2,4]. This type of network is an iterative gradient algorithm that is trained to minimize the mean-square error between its output and the desired result, as characterized by a set of exemplar data. Figure 3 depicts the general configuration of such a network.

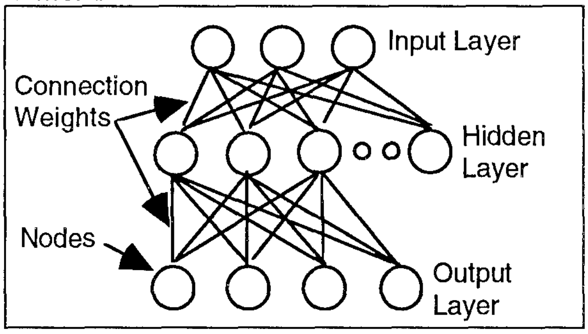


Figure 3: Typical Back-Propagation Network Configuration.

The number of input nodes corresponds to the number of spectral bands-per-pixel while the number of output nodes corresponds to the desired number of feature classes to be identified. The hidden layer allows the network to develop internal representations of the input data for re-mapping into the output classes. The number of hidden layer nodes is less deterministic, and depends greatly upon the nature of the input image data as discussed below. In Figure 3, each circle represent a processing node, with the connecting lines depicting the interconnection weights. Each node simply outputs a value which is dependent upon the sum of inputs. The connection weights (which are initially random) are adjusted during training until the output nodes generate a desired value in response to a known input.

To segment an image, the neural network is first trained on an exemplar set of pixels identified in the image. The exemplar set is identified manually by a photo-analyst, and includes pixels coded for a variety of different reflectivity features. Examples include bare soil, asphalt surfaces, cultural structures and water. The network is trained to a suitably low error value, then used to segment the entire image into the identified classes on a pixel-by pixel basis. To aid in the visualization of the segmented data, each feature class is assigned a pre-determined pixel value. The resulting image is inspected for systematic errors in classification, which are corrected manually. Minor errors in classification (such as the coding of a high-reflectivity pixel in the midst of an open field) are not removed. These artifacts become part of the background clutter that is present in a real-world scene.

In practice, we have used the segmented imagery to develop phototexture for radar processes in two different ways. Both depend upon the pre-assignment of values to pixels that represent displayed intensities of radar returns expected from typical grazing angles.

In our first approach, the segmented phototexture was used only to remove gross artifacts and adjust the underlying pixel values of the original phototexture. For example, water pixels would be set to a uniform low value (representing the return expected from a large forward scattering surface) while metallic structures are brightened. More recently, we have used the segmented image to define the geometric limits for Gaussian noise patterns with distributions appropriate for scatterers that display a Rayleigh distribution. The former method has the advantage of yielding a more detailed scene content, at the expense of inappropriate returns from some features. The latter method provides a more predictable, but less detailed return. For terrain processing, correlated DTED is used to provide underlying land form data.

Neural network application

Several different neural networks paradigms can be applied to process multi-spectral images to identify, on a pixel-by-pixel basis, the probable material class of the surface to support the processes outlined above. We have selected supervised networks such as the feed-forward back-propagation network described above over unsupervised clustering paradigms, although both have been applied to image segmentation[2]. Similar techniques are used in remote sensing to identify ore bodies and estimate crop yields from multi-spectral data. For the sample images here, a three layer back-propagation neural network was applied. The essential steps in the process are:

- extracting a suitable training set from the image
- configuring and training the neural network
- processing the raw image
- manual correction of classification errors

The sample image in Figure 4 is typical in that it contains a mixture of man-made structures and natural features. Through the segmentation process, each pixel is assigned to a pre-defined feature class. In the example here, the image was segmented into six classes, including soil, asphalt, concrete, man-made structures, vegetation, and water. The exemplar pixel groups used to construct the training set were identified through manual image analysis, with care taken to represent the variation present in the original image.

It is a frequent misconception that the exemplar set should consist of "perfect" examples of each class. Actually, the reverse is true. For example, the metallic structure exemplar pixel set includes both pixels for large industrial structures as well as for smaller buildings as the intent during processing was to assign both structures to a single class. The set of exemplar pixels for asphalt includes both pixels from taxiways as well as for the small secondary roads. In training a neural network, it is important to attempt to include exemplar pixels for the entire range of values expected for a particular class. This requirement is reduced in practice by corrupting the exemplar set data with random noise while the network is training. This reduces the tendency of the network to converge on a local minimum in the training data instead of a more general solution. Training with added random noise is continued until a recognition goal (typically set at 95% or greater RMS) is reached.

Figure 4: Sample phototexture

It is also beneficial to randomize the order of presentation in the training set, particularly where large numbers of identical, or nearly identical pixels are present in the exemplar set for a class. Without randomization, the network is easily trapped in a local minimum. This is indicated by tracking the identification error as each exemplar is presented to the network. An alternating pattern of high - low error as the exemplars are presented combined with an essentially fixed RMS error indicates a local minimum. Often, this is attacked by altering training parameters such as the percentage of random noise or reducing the training rate or "momentum" of the network. We have also found it useful to search the exemplars for identical pixels. Reducing the number of such pixels can correct local minimum problems. The risk in this approach is in altering the distribution of pixels in an exemplar set such that it is no longer representative of the input image.

It is helpful to view the exemplar sets as occupying one or more regions in a space defined by the number of spectral bands available. For the three-band data used here, it can be called RGB-space, and represented as a simple volume. For LANDSAT Thematic Mapper data, this space would be a seven dimensional structure. In either case, the exemplar set for a single class may occupy several disconnected regions, with other classes occupying intervening regions. The complexity of the image data in this structure requires the use of a network with one or more hidden layers in order for the network to re-map the input data into the desired set of output classes.

The optimum number of hidden layer nodes is a subject of some debate. Too few hidden layer nodes will prevent the net from converging, while too many adversely effects performance. Image data as a rule is complex, with several discrete groups of pixels in RGB-space assignable to a single class. Water for example, may vary from bright blue through green to brown depending upon suspended sediment, regions of RGB-space which can also contain soil, vegetation, and man-made structures. The number of hidden layer nodes is equal, in the worst case, to the number of disconnected or meshed regions in the input distribution[5]. As this is often a difficult number to determine, rules-of-thumb have evolved. Lippmann[5] states that there must typically be more than three-times the number of hidden layer nodes as input nodes. For the three-band data used here, a number of hidden layer nodes equal to the product of the input and outputs nodes is a conservative starting point.

After training, the network is used to segment the entire image on a pixel by pixel basis. As shown in Figure 5, the segmented image pixel values are assigned to a single value depending upon the feature class. In practice, this is a useful point at which to examine the segmented image in comparison to the original. While scattered errors are expected, systematic errors can usually be traced to errors in constructing the training set. Often, this can be resolved by noting those regions of the image incorrectly classified, and adjusting the exemplars in the training set accordingly. For example, regions of vegetation in shadow may be mis-classified as asphalt. By ensuring that exemplars for shadowed vegetation are included in the training set, a successful segmentation often results.

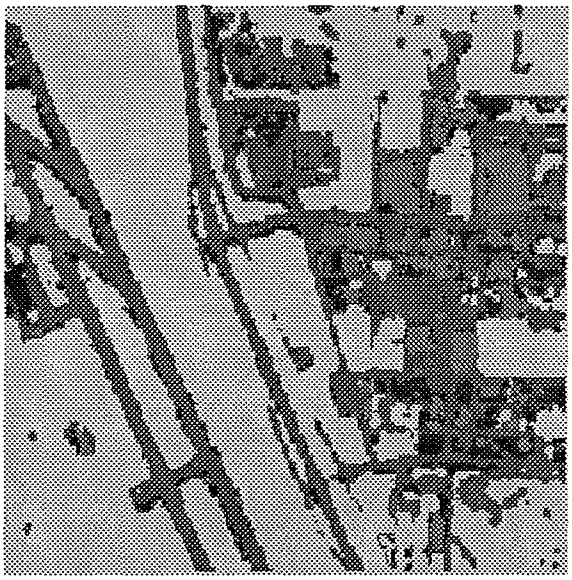


Figure 5: Example segmented image.

In other instances, the sets may not be separable. In our experience, beach sand is readily mis-classified as concrete. Close examination of the exemplar set values found that in the source image, many of the pixel values for the two surfaces are identical. This should not have been a surprise given that concrete contains a high percentage of local sand. A possible solution to this problem is obtaining source data with a larger number of spectral bands. To date however, we have elected to manually select and modify the critical mis-identified regions, which are readily detected in the segmented image. This is easily accomplished with commercial or public domain image manipulations tools.

While inseparable exemplars occur in natural imagery, a different problem exists in phototexture derived from visual system databases. In some cases, it may be desirable to use such phototexture as source data to avoid feature correlation issues. In such images, the total number of applied textures can be rather low, allowing nearly complete characterization in the training set. Also, the textures are typically applied in geometric patterns obtained from DFAD data. Although selected to avoid visual discontinuities, the different textures may be readily classified by a well trained neural network, resulting in a patchwork appearance in the segmented image due to distinct texture boundaries. In such cases, adjusting the original phototexture pixel values based on the segmentation data may produce unwanted edges. Our second approach to generating a database (i.e., re-assigning pixels in a given segment to a defined range of random values based upon the feature class) allows this problem to be reduced. For example, assigning separable mean points, but overlapping distribution limits, to soil and vegetation classes allows large regions to remain distinct while minimizing border edges.

While the approach described is applied to radar images in our prototype system, the database generation process can be extended to other sensor systems as well. For example, the segmentation value assigned to a given pixel may be selected to represent an infrared reflectivity and material heat capacity rather than X-band radar reflectivity. In a similar manner, databases for MMMW radar can also be constructed. Using this approach, highly detailed, geospecific sensor simulation can be produced to support world-wide missions.

Real-time processing overview

Real-time processing centers on adding aspect sensitive radar features such as leading edge enhancement and far-shore brightening to the image, and adjusting the gamma of the resulting image to that typical of a sensor display as shown in Figure 6. These processes are implemented using image processing algorithms that are aspect sensitive, allowing the approximation of the required radar effects. Also in real-time, DTED data is ray-traced to find both shadowed areas and forward scattering slopes. This information is combined with the processed phototexture to adjust the displayed values of the pixels. Additional processing necessary for DBS and real-beam map displays is not depicted in the figure.

Figure 6: Simulated SAR image using adjusted phototexture. Simulated illumination direction is from the northwest.

A generic form of cultural shadowing is used to produce occulted areas behind structures for which actual elevation data is unavailable. Generating this entails performing a rear-edge detection on a structure-only segment of the data base, and extending this edge as a function of grazing angle. While the images generated should not be confused with a rigorous radar signature prediction, they do provide efficiently computed, effective sensor simulation for training.

COMPUTATIONAL RESOURCES

In the prototype DRLMS system and in subsequent developments, the number of computations required for the generation of an image is directly scaled by the displayed range and azimuth resolution of the simulated radar system. Since image processing to produce representative SAR-like images replaces the more rigorous calculations of traditional DRLMS, the total number of computations required in real time are greatly reduced. For example, a tactical radar simulation may display an effective 256 x 256 resolution cells (the size of the simulated SAR image shown in Figure 3). The processes applied require on the order of 100 floating point operations per displayed pixel to transform the pre-processed data base into a representative SAR image (plus any additional overhead for the operating system, retrieving the database and performing interface processes). This translates to approximately 7 million floating point operations to build an image.

A modern RISC processor such as the MIPS R4000, operating at 100MHz, is rated at approximately 16 MFLOPS. Based upon the operations required, and neglecting the computing overhead previously mentioned, it is estimated that approximately 2 to 3 simulated SAR patches per second can be processed on a machine of this class. This is in good agreement with benchmark data from the Macintosh prototype, where processing times in the range of 15 seconds per patch are typical for a processor of 1/30th the rated floating point speed.

The benchmarks above predict the hosting of a real-time DRLMS on a single microprocessor. With newer generations of RISC processors such as the MIPS R4400 promising floating-point performance of 24 MFLOPS, further enhancements are achievable. This is in contrast to traditional DRLMS systems which depend upon special purpose hardware achieving hundreds of MFLOPS in parallel computing architectures.[6]

REFERENCES

1) Wehner, D.R. (1987) *High Resolution Radar* (PP 11-28) Norwood, MA: Artech House

2) Seldin, J.H & Cederquist, J.N. "Classification of Multispectral Data: A Comparison Between Neural Network and Classical Techniques" Proceedings of the Government Neural Network Applications Workshop: Volume 1 (PP 79-83) Wright-Patterson AFB, OH, 24-26 Aug. 1992.

3) Stengel, J.D. Jr. & Hoog, T.W. "DRLMS Technology- A Critical Assessment of the State-of-the-Art" Proceedings: 13th Interservice Training Systems and Education Conference (PP 35-43) Orlando, FL, 2-5 Dec., 1991.

4) Rumelhart, D.E. & McClelland, J.L. (1986) *Parallel Distributed Processing: Volume I: Foundations* (PP 318-362) Cambridge, MA: The MIT Press

5) Lippmann, R.P "An Introduction to Comuting with Neural Nets": IEEE ASSP Magazine (PP 4-22), April 1987.

6) Drew, E.W. & Matusof "Applying Advanced Parallel Processing Concepts to Radar Simulation and Image Generation" Proceedings: 15th Interservice Training Systems and Education Conference (PP 345-353) Orlando, FL, Nov. 29- Dec 2, 1993.

RAPID SIMULATION DATABASE BUILD USING HARDCOPY INPUT

Edward W. Quinn & Gregory S. DeLozier
Cartographic Systems Engineering, Loral Defense Systems
Akron, Ohio 44315

ABSTRACT

This paper describes the result of a research and development effort focused on developing technologies supporting rapid extraction of simulation databases. The specific goal of the project during this period of time was to significantly reduce the time required to extract features [roads, contours, streams, etc.] from graphic hardcopy sources [i.e, maps and charts]

This problem is significant to overall database construction cost and timelines. Currently, attempts are being made to use large maps scanners and commercial vectorization software to improve extraction efficiency. Unfortunately, the result of the use of only color or intensity to separate objects is that substantial interactive editing of the final product is necessary. This restricts the use of maps as an effective information source.

A new process was defined as a result of this task. It represents an integration of insights gained through the technologies of image processing, pattern recognition and neural network based learning. It represents two kinds of improvement: (1) A reduction in setup time [the operator need only identify typical objects, not define a complete color lookup table] and (2) reduction in interactive editing [by on the order of 90%] due to the higher quality of the output.

Examples are presented of images which illustrate the new process. They show the very significant capability which has resulted. In addition, possibilities for extension of the process to multi-spectral image data are defined.

BIOGRAPHIES

Edward Quinn is a Section Manager within the Cartographic Systems Department at Loral Defense Systems/Akron. His activities are focused on system design and implementation for database generation and database related products within the Simulation Division. He has written a number of technical papers, all on these topics. Ed holds a Masters Degree in Physics. In addition, Ed is an Adjunct Professor of Computer Science at Kent State University.

Gregory DeLozier is a Senior Scientific Analyst at Loral, specializing in scientific and technical applications of image processing, artificial intelligence and systems level applications of computer science. His current work is concerned with the use of integrated pattern recognition technology for cartographic database generation. He has written a number of technical papers in these fields. Greg holds a Masters Degree in Computer Science and is currently a Ph D. candidate. He is, in addition, an instructor at Kent State.

RAPID SIMULATION DATABASE BUILD USING HARDCOPY INPUT

Edward W. Quinn & Gregory S. DeLozier
Cartographic Systems Engineering, Loral Defense Systems
Akron, Ohio 44315

INTRODUCTION

The rapid construction of databases is a long term goal which supports the objectives of all phases of simulation. The criticality of this technology is particularly important for applications such as mission rehearsal. The cost reductions which the technology implies are of increasing importance as well, due to the increasing proportion of total system costs which are associated with database construction.

For simulation based on the presentation of real-world terrain, a problem is that a great deal of the geographic knowledge of the world remains stored on hardcopy maps and charts. Paper maps represent an intellectual resource which has been refined by geographers over several hundred years as a carrier of information in a format which can be readily assimilated by human beings.

Although attempts to automate information extraction from paper maps are continuing, it is estimated that roughly 90% of the potential information which could usefully be extracted remains only in hardcopy form.

This difficulty is due to the very different methods which computers [and image generators] require that geographic data be stored. An illustration of this difference is the fact that the only use of map-like hardcopy graphics in a simulation environment is to convey the content of the simulation database to human beings.

This mismatch between the way that human beings have represented and interpret data and the very specialized formats required for real-time simulation is a fundamental problem which impedes advances in the use of simulation as a tool to refine designs, train equipment operation and aid the planning process, regardless of the application. Rapid database build is the technology which attempts to bridge this difference and resolve the problem in a cost effective way.

TECHNOLOGY BACKGROUND

1. Data Tablets

Initial efforts to extract information from paper maps were focused on efficient methods of using data tablets to manually delineate the geographic positions of objects, using a special cursor control device [a puck] to do so. For this task, the puck was augmented to include function keys and even miniature numeric keyboards. By this means the necessity of the operator of continually shifting their attention from the computer keyboard to the puck was minimized.

Other techniques, such as staged data extraction and efficient editing tools were also introduced. These allowed the complete data collection and attribution task to be achieved at an overall average rate of approximately one feature [lines, points or areas] per minute.

For specialized tasks, pucks which included image sensor arrays were also introduced. For high contrast [black/white] idealized sources, these reduced the necessity of the operator to precisely follow the line. These restrictions on the use of this technology prevented its widespread application.

2. Softcopy Digitization

Hardcopy image scanners became available, initially as a result of process color industry applications. These are capable of rapidly converting the hardcopy map to a digital image. The increasing power and display capability of the computer workstation, when coupled with the recent introduction of low cost desktop scanners, has given rise to increasing use of softcopy [or "heads-up"] digitizing. The capability to control all modes of the process [editing, data collection, merging, database management, etc.] directly from the workstation's graphic screen has made this mode of operation increasingly popular.

3. Spectral/Intensity Extraction

Given the widespread availability of image scanners, it is natural to attempt to directly use the image as the information source for extraction [i.e, without human interaction]. In viewing a hardcopy map, it [at first glance] seems obvious that the use of discrete rather than variable colors, standardized symbology and stable scaling would make a color based automated extraction process relatively simple. Indeed, a number of commercial ventures are based on exactly this approach. Such a process is illustrated [at the top level] in Figure 1.

Unfortunately, complications exist which prevent the complete realization of the objective of a highly automated approach. These include such factors as the use of halftones [in which discrete color combinations are used to represent intermediate colors] and the physical size of the scanning aperture [which mixes colors over a wide range at the boundaries of objects]. As a result, the color map image is presented to the computer as a series of pixels with color values ranging over a wide scale. No convenient process is available to convert the digital image to a form more nearly representing that which is perceived by the human eye.

A consequence of this limitation is that errors are incurred in extracting separate 1 bit image layers for each object type. This results in the requirement for substantial manual intervention and editing operations. Only approximately a 2:1 improvement is achieved in efficiency [over puck based digitization]. Given scanner and software acquisition costs, as well as the cost of editing itself, this prevents the widespread use of this process for converting color map information to an intelligent vector database.

IMPROVED PROCESS

Our organization has explored this problem in detail for both maps and multi-spectral imagery for several years. This focus is due to the importance of this technology to efficient, rapid extraction of information for mission rehearsal, planning and other simulation applications.

In April, 1993 a new initiative began which examined the issue from the perspective of an integrated approach. It was observed that a number of attempts had been made in the past which attempted to solve the color extraction problem. Each had examined the problem from the perspective of the potential gain to be realized from the use of a single element of technology.

An example of this is the use of statistically based extraction, which attempts to derive optimal feature class boundaries in color space. A variety of algorithms [K-Means, Isodata, for example] are available for such feature class partitioning. However, as mentioned previously, the use of halftones on maps and a finite scanner aperture size significantly degrades the result of any color extraction process. The colors presented to the computational process are not completely unique.

Hence a technique which is based only on color is guaranteed to result in errors.

Before beginning the process of researching an integrated approach, the following objectives were established:

1. The new process would easily interface with current methods [color based extraction], without materially affecting overall process flow. This would ensure a relatively easy transition to new methods.

2. Operator interface/training would simplify the process rather than increasing its complexity. In essence, it is required that the operator be concerned with identification functions rather than functions based on setting up a data structure.

3. The development process would be such that prototypes could be easily constructed, strung together, and revised [or expanded] to handle additional classes of problems [such as multi-spectral image extraction].

The above objectives were achieved as follows:

1. Process Interfacing: The new process accepts the same 24 bit color raster input file currently being used for feature extraction. The output is a series of 1

bit/pixel images which indicate the presence or absence of the desired object. These files are directly processed by commercial vectorizers, just as current outputs are.

2. Operator Interfacing: The use of a neural network to interactively develop classification parameters was chosen. Rather than directly setup the process pipeline, the operator identifies examples of object types. The network then setups the pipeline with the proper parameters, after it has "learned" them. This identification process is simpler than the process of attempting to setup a color look up table.

3. Development Environment: Use of the Oberon Object Oriented Development Environment [1] was chosen to allow rapid prototyping within an environment that could support an efficient visual user interface.

In addition, Oberon allows process insertion and modification to be accomplished interactively without long recompilation sequences. Recompilation of even procedures that are thousands of lines of code consumes only seconds of time and can be performed without recompiling unaffected portions of the code body.

An additional assumed requirement of the developed process was that, in order to maximize the probability of success, a multi-stage classification approach would be used. As finally implemented, the process uses mixtures of trained networks and conventional pattern recognition and/or image processing algorithms to solve specific components of the problem.

The basic principle of our process relies on a coarse to fine classification. The advantage of this approach is that for simple initial decisions, only a subset of the total possible parameter space need be used, leading to rapid decisions as well as easy learning and training evaluation. As the process continues, the tests become more refined and selective. Consequently, the use of specific image measurements is injected in order to clearly differentiate between closely related possibilities. By minimizing the range of possibilities at each stage and restricting parametric inputs to only those of significance, processing time may be constrained to be within acceptable limits and the complexity of the process at each stage is constrained as well.

At the end of the process, a most probable classification has been assigned to the pixel being tested. The appropriate location is marked in the corresponding feature separate layer [one layer for each type, such as road, contour, etc.]. For cases where a strong degree of ambiguity still exists, two locations may be marked.

This redundancy is also a capability of the process that is not available with conventional approaches. It significantly reduces the possibility of gaps and holes in the output [which must be filled, either interactively or automatically]. As a result, image postprocessing to correct such deficiencies [prior to vectorization] are significantly reduced and simplified.

As with other elements within the processing pipeline, this kind of
processing is also adjusted to be sensitive to feature type.

PROCESS ILLUSTRATION

In order to demonstrate this procedure, a simple example will be used. This example concerns the extraction of contour information from a map. Such a process would be required in order to derive a terrain surface for simulation [with additional processing of the contours for actual generation of the surface].

Figure 2 shows the original map. It is a standard color map scanned using a conventional 300 DPI color desktop scanner. The output file being viewed may be formatted as a 24 bit TIFF file, or in variety of other formats for viewing.

When viewing the image, the operator can identify on the large screen area devoted to the map a number of examples of each kind of feature [Figure 3]. A maximum of eight object types can be simultaneously trained. After initial samples [and counter examples] have been collected, the pipeline solution to classification can be generated.

The operator can then probe the solution by using the cursor to designate the center of a small square region and then examine the solution [several of these are present in figure 4]. If in error, the samples are immediately corrected [relabeled]. After a suitable number have been examined, the process cab be reiterated if necessary.

Experience has shown that a maximum of three passes of the solution/correction cycle is necessary. The process has been tested by both persons familiar with the algorithms involved and with relatively unfamiliar personnel, with similar results.

A single bit image showing the result of classification is shown in Figure 5. The result of the new process results in very stable line work and little noise. This significantly reduces the interactive cleanup work which would otherwise be necessary before attempting vectorization.

Two additional stages of processing are shown in Figures 6 and 7. In Figure 6, the contours have been vectorized and labeled. This process involved a small amount of interactive time to review the data [always desirable during data collection] and then attribute it. This time was much less, however, than would have been required with the color based data.

Figure 7 shows the resultant terrain surface in a perspective view. The contours were converted to a Triangulated Irregular Network ["TIN"] and then a terrain matrix constructed.

FUTURE WORK

At this point, the process is nearing the point where it may be used as an operational capability. It has been demonstrated to work with DMA Arc Digitized Raster Graphic [ADRG] images, images from large format scanners, and desktop scanners of a variety of qualities. Although work could still be performed to enhance the user interface and further reduce processing time, the advantages of the process are readily available at the current stage of development.

The success of the technique is also encouraging our group to apply it to multi-spectral imagery. The problems of map data collection bear much more resemblance to image processing than we had first imagined when the project was initiated.

A similar success with imagery would significantly enhance data collection for areas where current hardcopy map data of the proper scale is not available. It appears to us that by tuning the spatial descriptors to reflect the scale of objects which must be extracted, the outline of the current process can be largely kept intact. Here, as with the map data extraction process, the goal would not be to solve all possible classes of feature extraction problems at once. Instead, it would be to significantly enhance and simplify current methods.

It is our belief that the availability of this approach would be of great value to the problem of extracting databases for simulation and other applications.

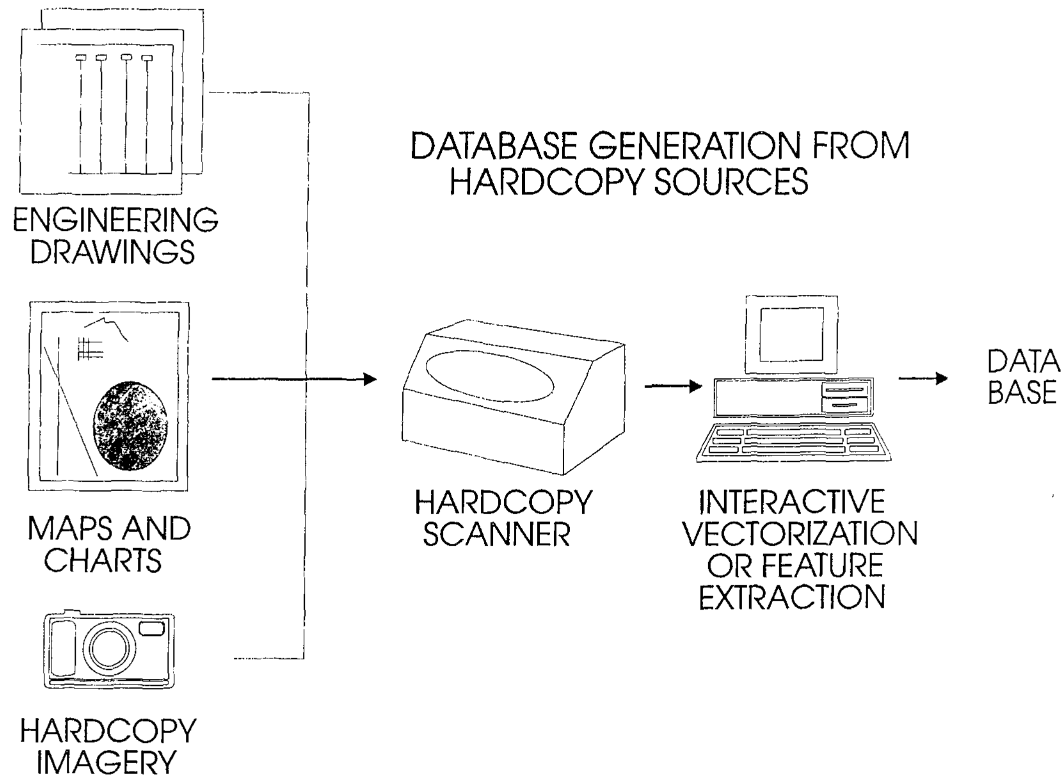


Figure 1 Hardcopy Processing Sequence

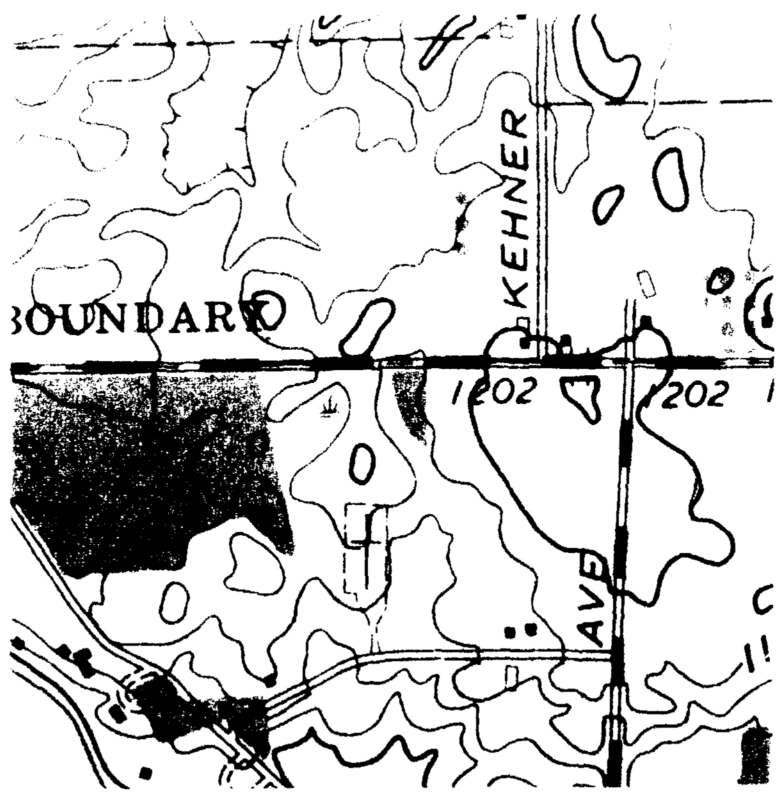


Figure 2 Original Color Map

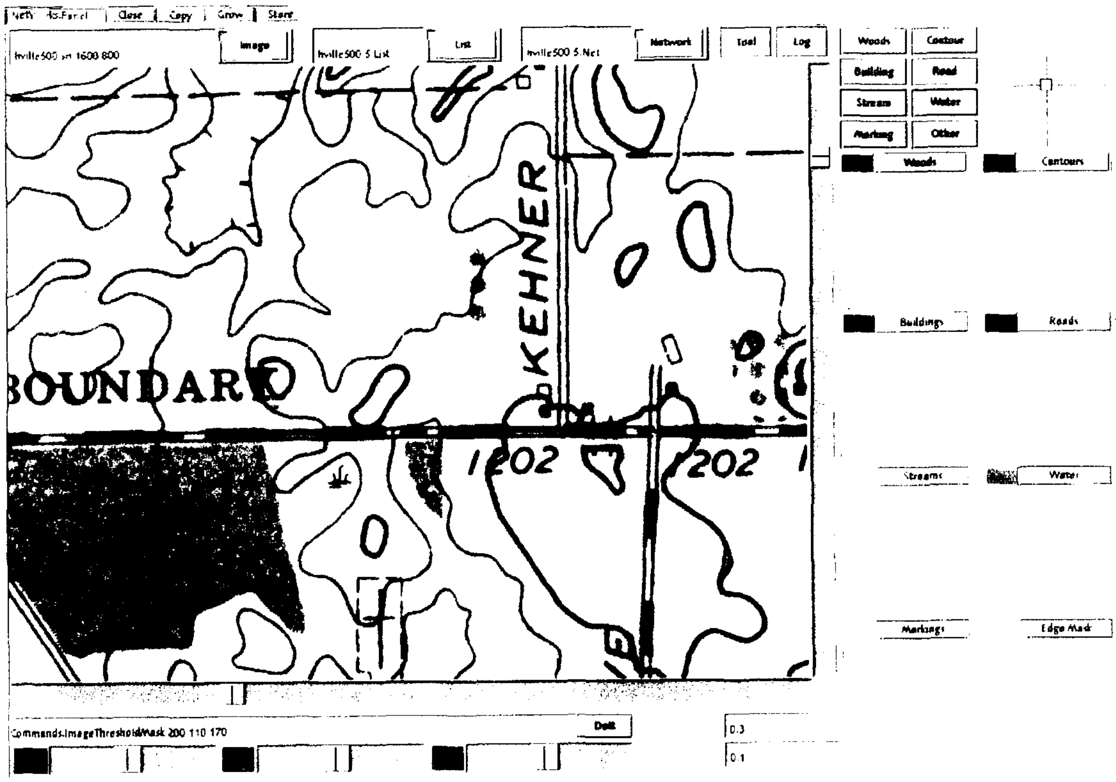


Figure 3 Color Map Interactive Screen

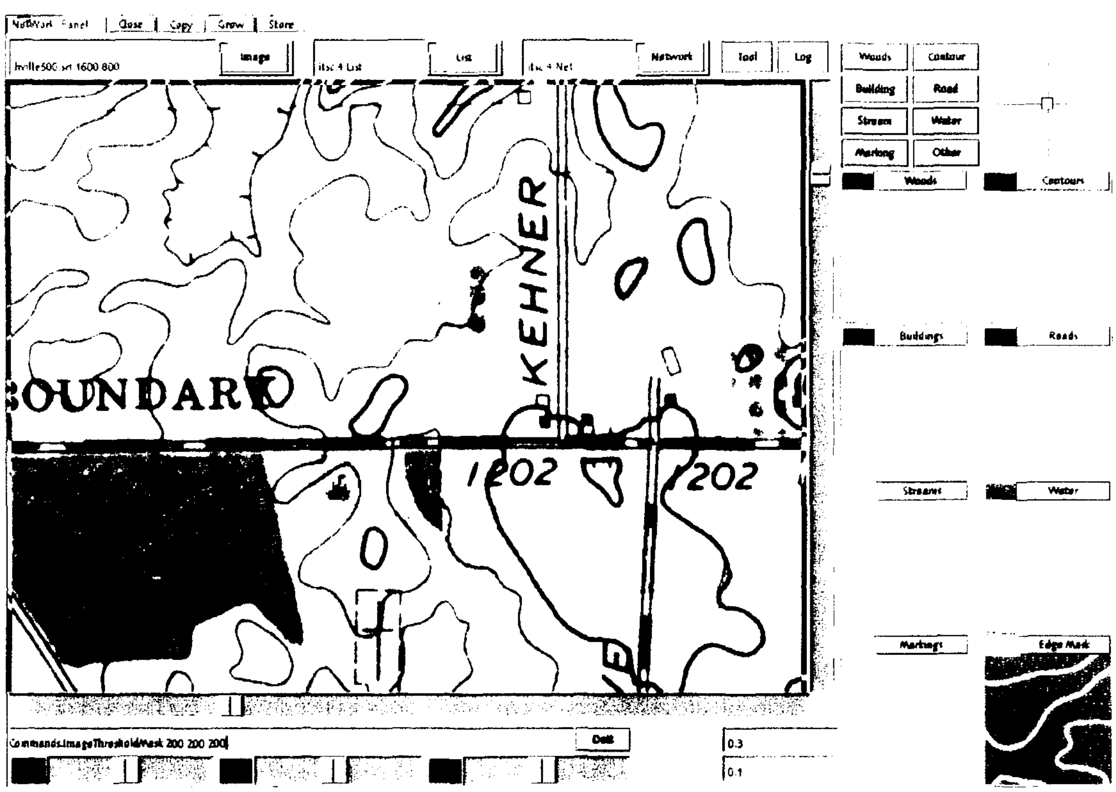


Figure 4 Trial Classification for Map Output

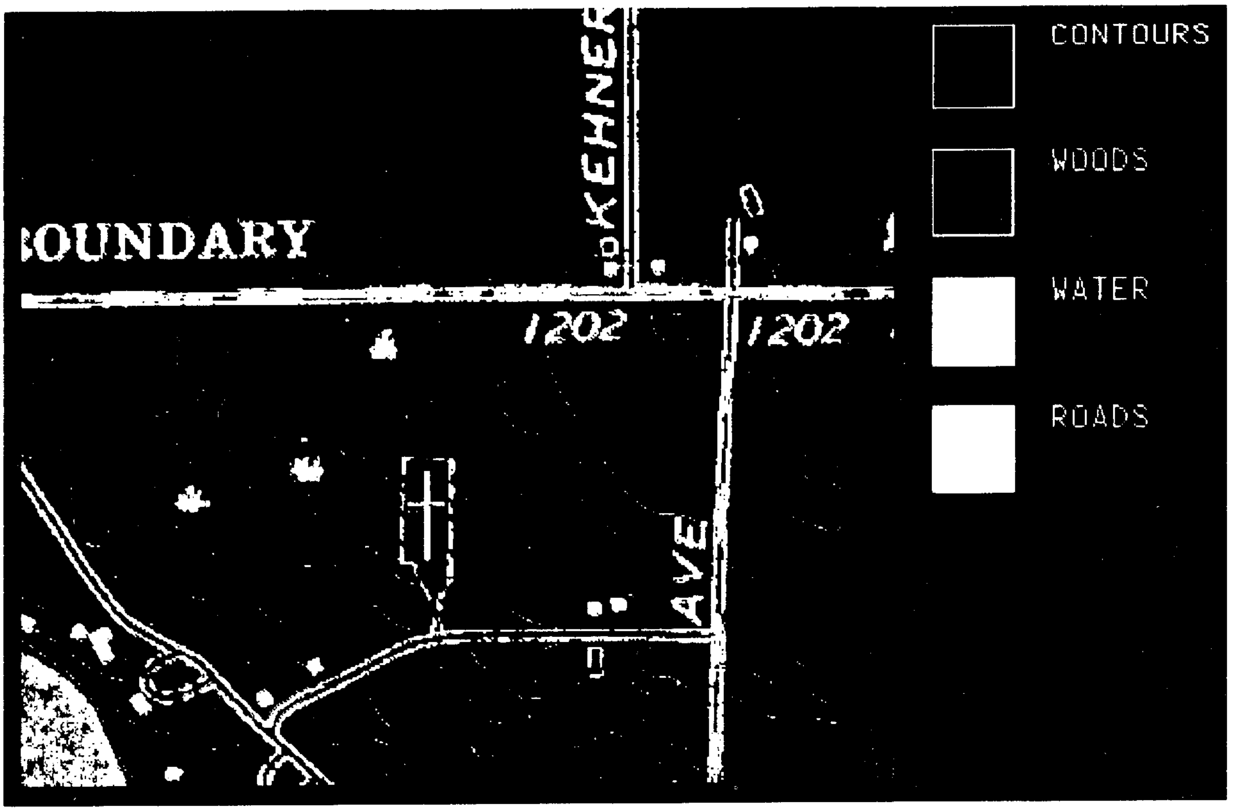


Figure 5 Classified Map Output

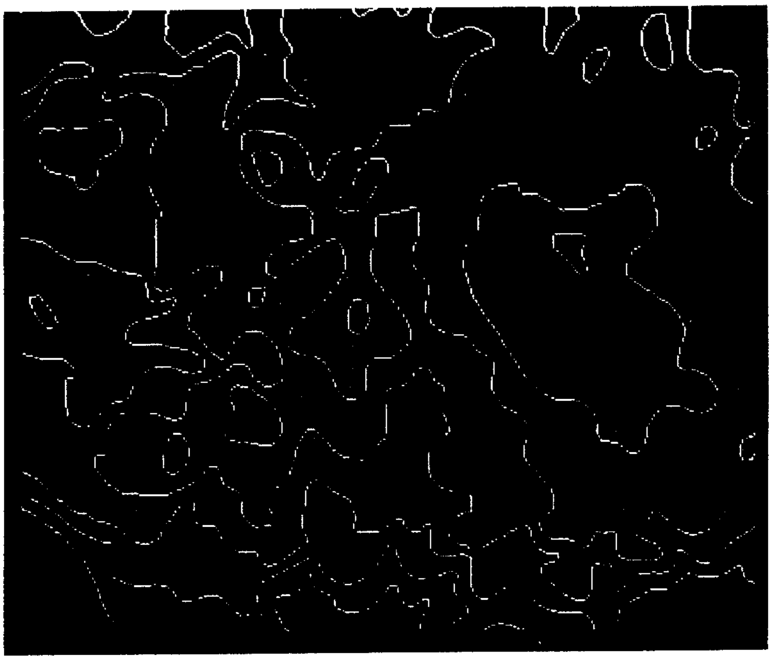


Figure 6 Map Vectors for Contour Layer

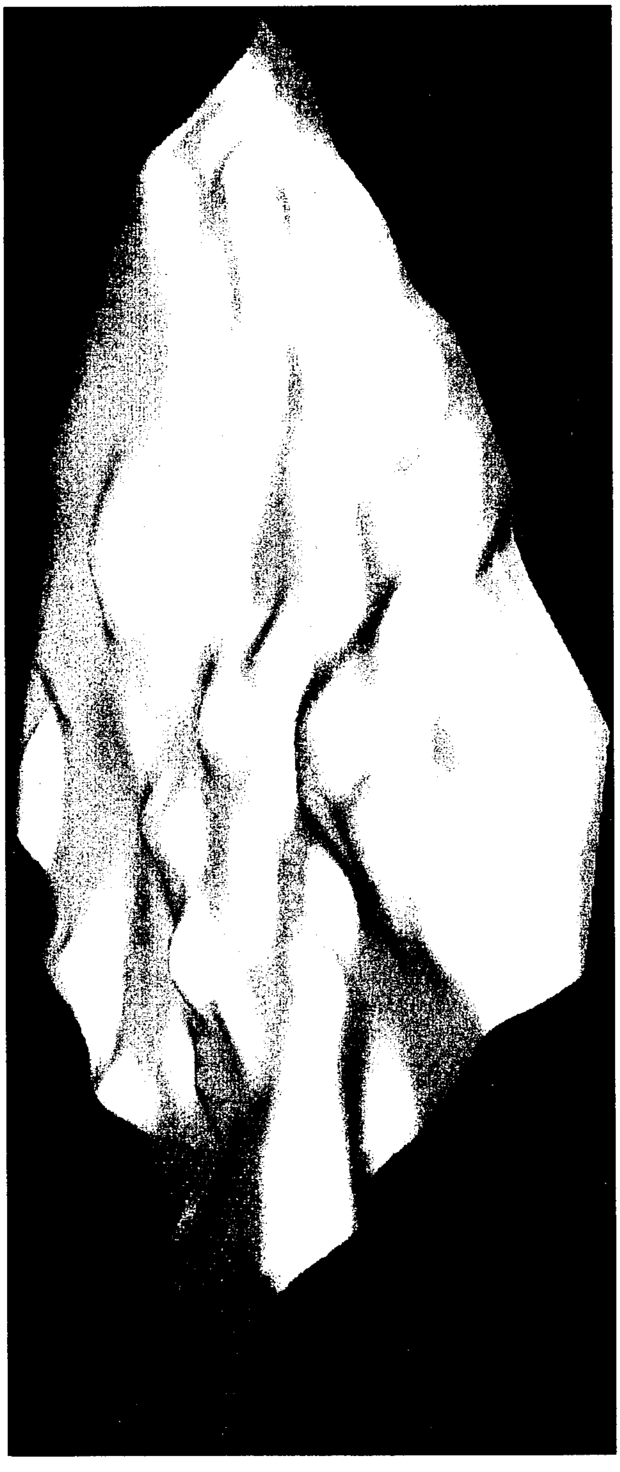


Figure 7 Perspective View Using Map Contours

SMART JARMS – COMPUTATIONAL INTELLIGENCE IN SIMULATION

Steve Manzi, Shelia Burgess and Debbie Berry
Martin Marietta Information Systems Company
Flight Systems Group
Orlando, Florida

INTRODUCTION

The ability to simulate Electronic Combat (EC) is a vital part of networked virtual reality simulation. At Kirtland Air Force Base in Albuquerque, New Mexico, the 542nd Combat Crew Training Wing utilizes this capability to support Special Operations Forces mission rehearsal and training. Importance to National Command Authorities of successful missions rehearsed at this facility cannot be overstated. Therefore, the highest level of fidelity to the real world must be achieved during simulation.

Pursuing an upgrade to the EC simulation as proposed herein would enhance overall mission rehearsal capability. EC simulation consists of software developed in the late 1970s rehosted on modern hardware. Although adequate, improvements are possible. This study examines the EC simulation to determine where computational intelligence techniques can be applied to provide an improved solution.

The purpose of this study is to investigate two Computational Intelligence (CI) candidates for replacement of the Offensive Tactics Simulation. The two techniques employed are Fuzzy Systems (FSs) and Neural Networks (NNs). FSs are knowledge based systems that allow subjective manipulation of inexact concepts. NNs are an idealization of the interconnections and functions of a nervous system which mimic the brain's learning and thought processes. Initial modeling of both techniques is performed and the results reported. Implementation of either system could potentially yield performance improvements.

Overviews of real world weapon systems and the current EC simulation are provided. Then, the approach used to develop CI solutions is defined. The FS and NN solutions are examined along with implementation considerations, empirical results, and conclusions.

Weapon System Overview

Modern weapon systems are usually composed of many subsystems, three of which could be a sensor to detect targets, a guidance system to direct the weapon, and the weapon itself.

A typical example would be a radar guided surface–to–air missile (SAM) system. Radar subsystems typically exhibit a number of distinct modes which are differentiated by observable characteristics, some of which are Radio Frequency (RF), Pulse Width (PW), Pulse Repetition Frequency (PRF), and Scan Pattern. The overall weapon system state would progress from mode to mode based on specific parametrics of the ownship/weapon engagement. Some of these parametrics are:

- Range
- Altitude
- Elevation
- Occult Status
- Jamming Effectivity
- Weapon in Flight

For example, a radar guided SAM could typically progress through the modes shown (see Figure 1) during an inbound radial ownship/weapon system engagement.

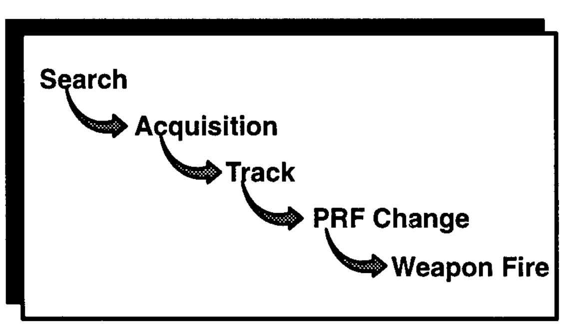


Figure 1. Mode Progression

Individual weapon systems may be joined into Command, Control and Communications (C^3) networks which affect their response. Under the conditions of open conflict the response time of individual weapon systems is typically decreased. This occurs when the search phase of the engagement is made more efficient when data from early warning systems are made available to fire control systems. Under different circumstances, components of the chain of command may exercise control over a specific encounter for political or military reasons, thus response time could increase.

Current EC Simulation

The current simulation of weapons systems is via a pseudo object oriented approach in which software entities known as JARMS (Jammer, Artillery, Radar, Missile Systems) simulate weapon systems. The instantaneous state of the overall simulation is contained in a file known as the Master Electronic Warfare Environment (MEWE). Among other data, the MEWE contains position and mode of all pertinent EC entities.

In order to mimic the mode changes of real–world threat systems, the JARMSs monitor the environment via the MEWE and extract variables which are manipulated by a pre–determined set of rules. These rules, referred to as Tactics Algorithms (TAs), are tailored to the specific modes of each JARMS. A unique combination of tailored TAs define a specific JARMS behavior. The current EC simulation utilizes 28 TA templates.

The MEWE is scanned and TAs are processed in an iterative fashion to drive JARMS into different operational modes until termination of the engagement. Termination may occur due to range, target occult, break lock, or target kill.

Real world weapon systems may exhibit many distinct modes. However, JARMS have a maximum of eight, one inactive (off) and up to seven active. Observable characteristics of simulated modes are designed to correlate to selected modes of the real–world weapon system, to the limit of the fidelity of the system.These characteristics are made available, again via the MEWE, to the portion of the application that simulates the EC suite components onboard the ownships; i.e., radar warning receivers, electronic counter measure jammers, etc. These models attempt to detect and/or counter JARMS. This creates a closed–loop interaction between the JARMS and simulated EC suite components.

As shown (see Figure 2), TAs, and therefore mode changes, are a function of the following environmental inputs:

a. Ownship to JARMS range

b. Ownship altitude

c. Ownship elevation

d. Electronic Counter Measures (ECM) Effectivity

e. Weapon in Flight Status

f. Ownship to JARMS delta altitude (for Airborne Interceptors)

g. Ownship occult status

APPROACH

To provide a proof–of–concept that CI techniques are viable solutions to EC simulation, FS and NN "drop–in" replacements were designed that will respond to the same inputs and produce the same outputs as the current TAs, but with higher fidelity.

Envisioned advantages of a CI implementation include:

- New JARMS tactics are created and verified by a labor and time intensive trial and error approach. The trainability characteristic of NNs and the intuitive properties of FSs could reduce this effort.
- Processing of other information which is currently available, but not used, could be added to enhance the fidelity of the simulation. An example of such data is aspect of ownship to weapon relative bearing, an important real world parameter. Either a FS or NN implementation could process this data potentially creating a higher fidelity simulation than what currently exists.

To bound the problem, occult status was not used and only SAM type JARMS were investigated. Terrain masking data could be added later.

Due to sensitivity associated with EC applications, a fictitious SAM JARMS designated as the SA–A has

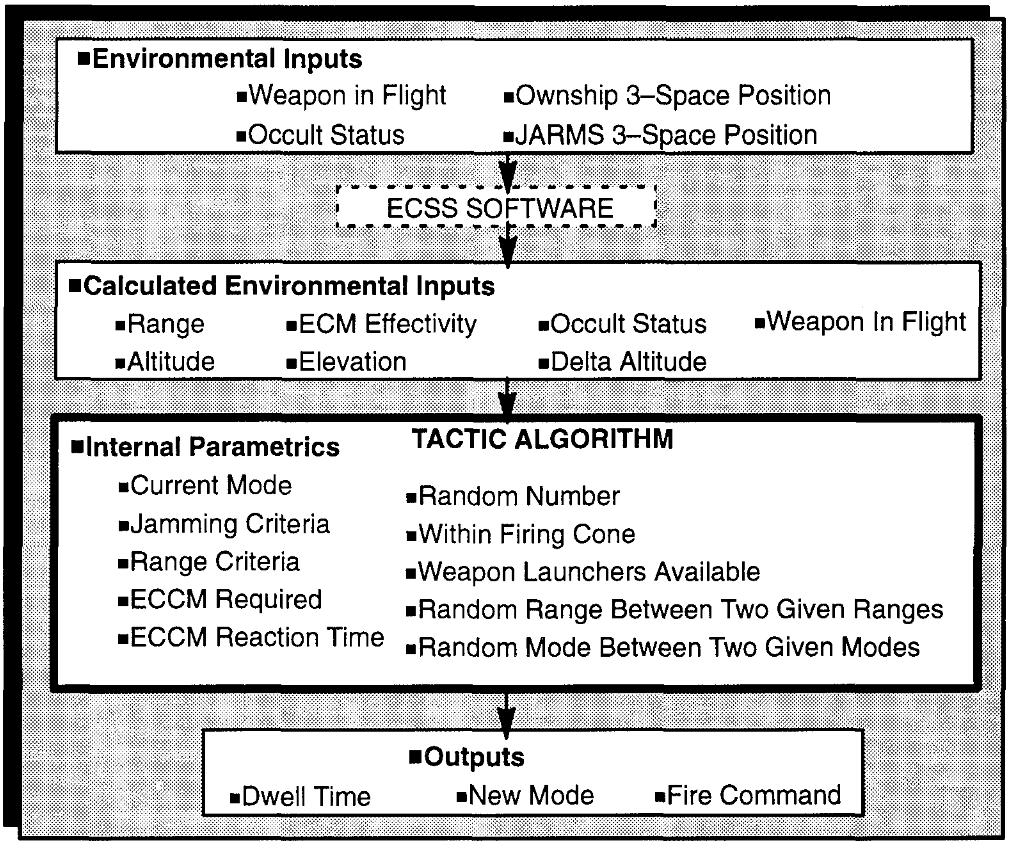


Figure 2. Existing Implementation Process Flow

been created. The SA–A utilizes parameters specified in the following TA definitions to model its behavior:

Current Mode of JARM = 0 (Off)
If Ownship Range > 90 nm
Then dwell in mode 1 for 60 sec
If Ownship Range < 90 nm
Then dwell in mode 2 for 30 sec

Current Mode of JARM = 1 (Search)
If Ownship Range > 65 nm
Then dwell in mode 2 for 60 sec
If Ownship Range < 65 nm
Then dwell in mode 1 for 30 sec

Current Mode of JARM = 2 (Acquisition)
If Ownship Range < 45 nm
If Jamming Effectivity > 40%
If Ownship Range > 35 nm
Then dwell in mode 3 for 20 sec
If Ownship Range < 35 nm
Then dwell in mode 4 for 20 sec
If Jamming Effectivity < 40%
Then dwell in mode 5 for 10 sec
If Ownship Range > 45 nm
Then dwell in mode 1 for 60 sec

Current Mode of JARM = 3 (Acquisition)
If ECCM is required
Then dwell in mode 4 for 20 sec
Else If Ownship Range > Previous Range
Then dwell in mode 1 for 60 sec

Current Mode of JARM = 4 (Low PRF)
If Ownship Range < 25 nm
If Jamming Effectivity > 50%
Then dwell in mode 3 for 20 sec
If Jamming Effectivity < 50%
Then dwell in mode 5 for 20 sec

Current Mode of JARM = 5 (High PRF)
If ECCM is required

Then dwell in mode 4 for 20 sec
If ECCM is not required
If Ownship Range > 25 nm
Then dwell in mode 3 for 20 sec
If Ownship Range < 27 nm
and Altitude > 500 ft
and Elevation < 70 deg
and Jamming Effectivity < 25%
Then Fire Weapon

Fuzzy System Solution

The Fuzzy Inference Process is the algorithmic process internal to the FS that provides the ability to manipulate abstract concepts in a fashion similar to human decision making. Fuzzy Systems need to interface with non–fuzzy applications, so the fuzzy process must accept and output crisp (mono–valued) data.

The Process is separated into three subfunctions:

1. "Fuzzification", where a crisp input value is transformed into a fuzzy value.
2. Rule Evaluation, where the fuzzy output truth values are computed.
3. "Defuzzification", where the computed fuzzy value is transformed into a crisp output value suitable for non–fuzzy applications.

Fuzzification is accomplished by applying crisp inputs to more than one function to produce a set of intermediate outputs. The shape and overlap of the functions in the domain of the range of the input provides ambiguity in the intermediate output set. For example, the FS range implementation for the SA–A used overlapping linear functions as shown (see Figure 3). The input data set is applied to previously developed rules, examples of which are shown (see Table 1).

The other input variables modeled in the same manner as the range include:

- Mode
- Jamming Effectivity
- ECCM Required
- Reaction Time
- Altitude
- Elevation

Each single input is transformed into one or more outputs. In the example provided, a range input of 45 nautical miles corresponds to the range membership function outputs of:

- 0 % Very Far
- 0 % Far
- 60 % MidNear
- 30 % Mid
- 0 % Near
- 0 % Very Near

A jamming effectivity of 53% corresponds to a jamming effectivity membership function output of:

- 0 % Very High
- 0 % High
- 100 % Mid
- 70 % Low Mid
- 0 % Low
- 0 % Very Low

During processing of the FS, all rules are exercised at the same time so, all these input variables are fuzzified as part of the system. An example of processing the two rules is given below:

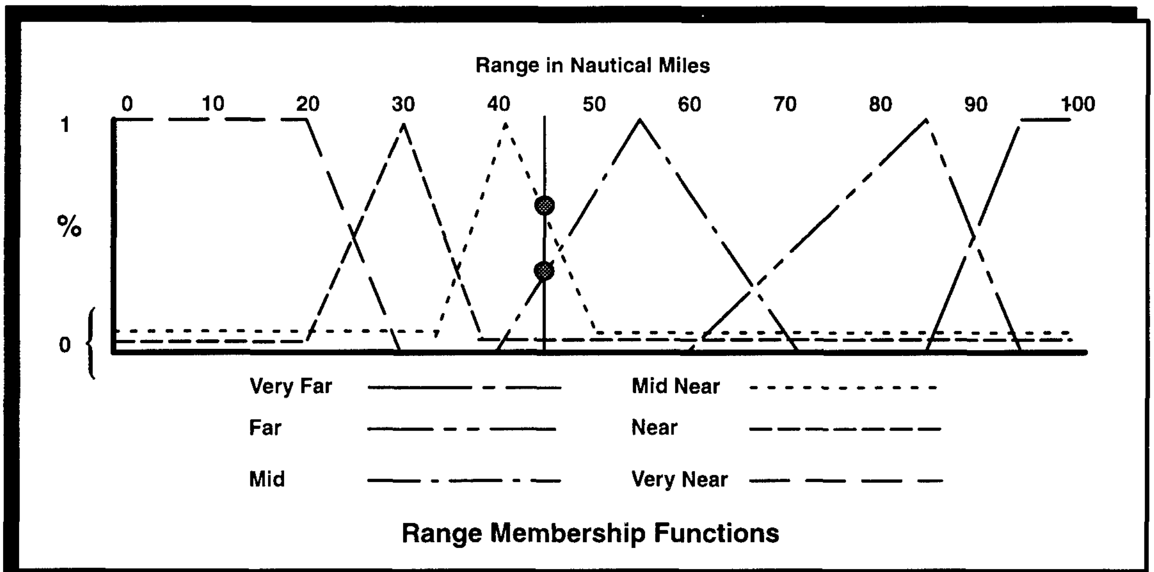
1. As shown (see FIgure 4), for each rule a product equivalent to the logical "and" of the inputs would be taken, in this case 60% for range and 100% for jamming effectivity. The logical "and" of these two inputs is the minimum, or 60%. It can be seen that the product of those sets will be zero for each input that results in a membership output of zero.

2. Next a product equivalent to the logical "or" of the resultant output of all the rules is taken. This corresponds to the superposition of the logically true portions of individual rules.

3. Outputs of these rules are input to an averaging function that uses centroid calculations on the resulting structures to yield crisp outputs. The crisp output values are input to the EC Simulation processes requiring control data of JARMS behavior.

The utility of the FS approach was immediately obvious. The FS used to model the SA–A was quick to construct, easy to debug and produced acceptable performance.

FS is implemented in ANSI C code that would be straightforward to add to the existing simulation.



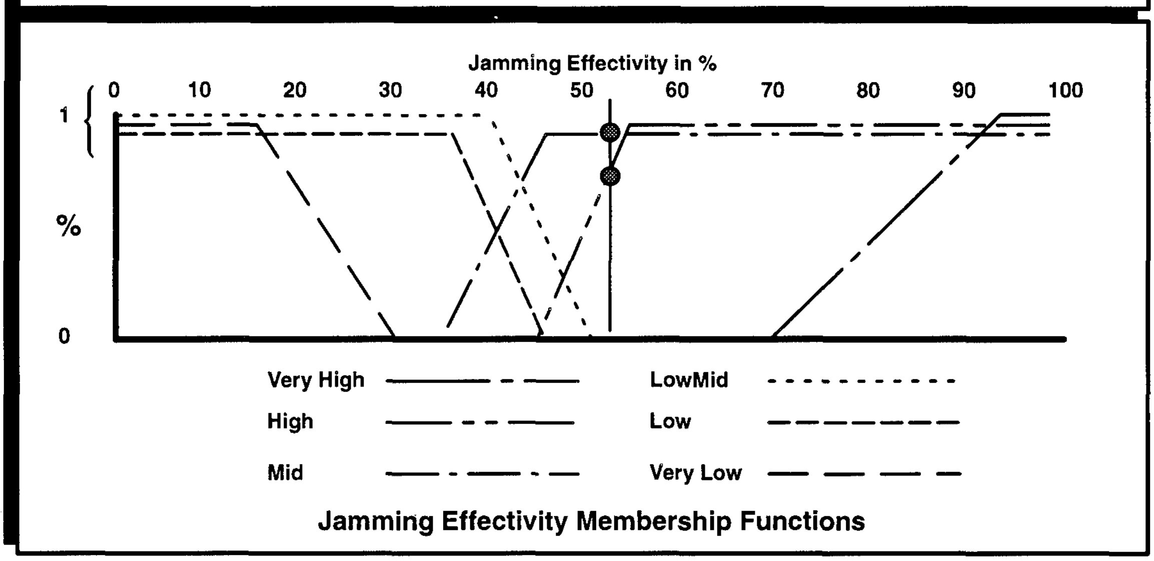


Figure 3. Membership Functions

There would be no hardware impact. The software impact would entail modifications to the executive/scheduler for inclusion of the new modules. The C source code would be cross complied on the development system and the resulting object code added to the simulation build.

Neural Network Solution

Neural nets provide a means of correlating data using simple processing elements. Basic neural net building blocks are neural units, analogous to the biological neuron. A neural unit accepts inputs and applies weights to them. These weights are the analog of biological synapses that connect neurons. Weighted inputs are summed and applied to a simple, predefined function. The result of this function is transmitted as the neural unit output. Units are associated into layers with the overall net being classifiable based on the number of layers.

Three layer neural nets have gained wide acceptance due to their versatility and ability to produce acceptable outputs with a minimum number of layers and their associated learning time overhead. These nets group units into an input layer, a hidden layer, and an output layer. The association of units in a three layer net is depicted (see Figure 5). This network represents the inputs to and outputs from one JARM. Each discrete input is associated with an input node, and each discrete output is associated with an output node. The number of nodes in the hidden layer are directly

1. if range is MidNear and jam_eff is Mid then new_mode is Acq_ECCM
2. if range is Very Near and jam_eff is Very Low then new_mode is High_PRF

1. if current_mode is Activate and range is VeryFar then new_mode is Search
2. if current_mode is Activate and range is Far then new_mode is Acquisition
3. if current_mode is Search and range is Far then new_mode is Search
4. if current_mode is Search and range is Mid then new_mode is Acquisition
5. if current_mode is Acquisition and range is MidNear and jam_eff is Mid and range is Near then new_mode is Low_PRF
6. if current_mode is Acquisition and range is MidNear and jam_eff is Low then new_mode is High_PRF

Table 1. Samples of FS Rules

proportional to the number of cases the neural net must learn and desired precision of output data.

NNs form the transfer functions that map input data to output data. Neural nets are trained, not programmed. The training process provides inputs to the net for which there exist known outputs. The collection of weighting factors (sometimes referred to as the gain matrix) between the elements of the net are then iteratively perturbed via the back–propagation algorithm so as to drive the error in the output to a predetermined value. Once trained, a network is ready to be tested by accepting non–training data. For each new input case, the network produces a best estimate output with a corresponding measure of uncertainty.

Speed is a primary advantage over conventional processing. Neural nets make connections almost instantaneously and processes data in parallel. Also NNs can produce answers based on incomplete input data sets.

Upon examination of this application and the TAs, several NN implementations were investigated. In the first implementation, a NN configuration was developed that attempted to accommodate all SAM type JARMS. This implementation used a NN for each JARMS TA/mode.

The goal of the first implementation was to develop a set of NNs consisting of one NN for each of eight TAs used. All eight NNs used the same input and output nodal configurations. However, these nets were trained with different case sets of data. The unique case sets were developed to support each TA's behavior the network was to learn. This approach could then permit an operator to develop new SAM JARMS neural nets by simply creating appropriate training and test case sets.

The resulting NN consisted of 6 inputs and 3 outputs as shown below.

Inputs:

- JARM Identification Code
- Current Mode
- Range
- Jamming Effectivity
- Altitude
- Elevation

Outputs:

- Next Mode
- Dwell Time
- Fire Weapon Command

A neural network of this configuration was then constructed for the 6 TAs of the SA–A. Training cases were developed and used to teach the networks the SA–A's behavior. These networks

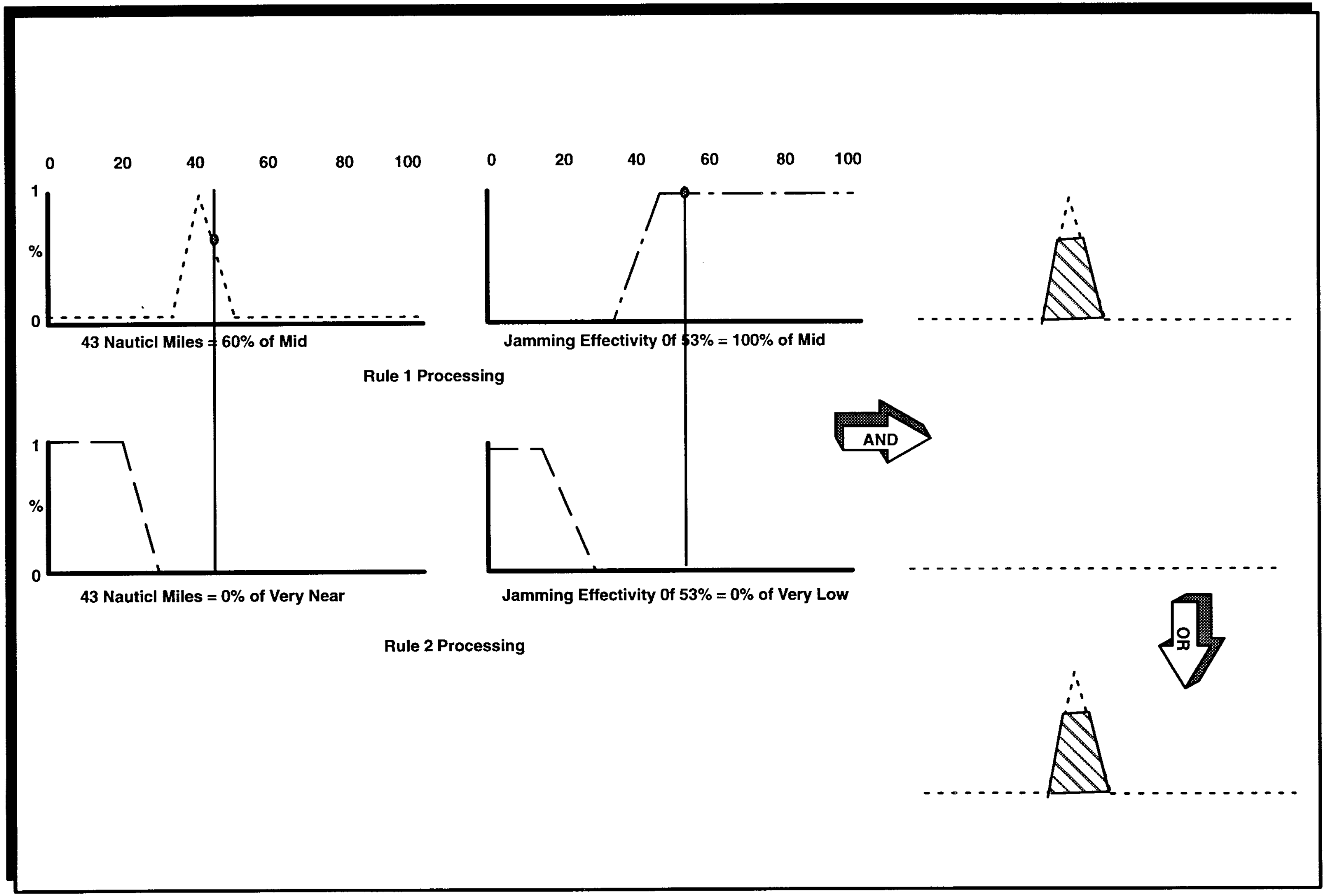


Figure 4. Rule Processing

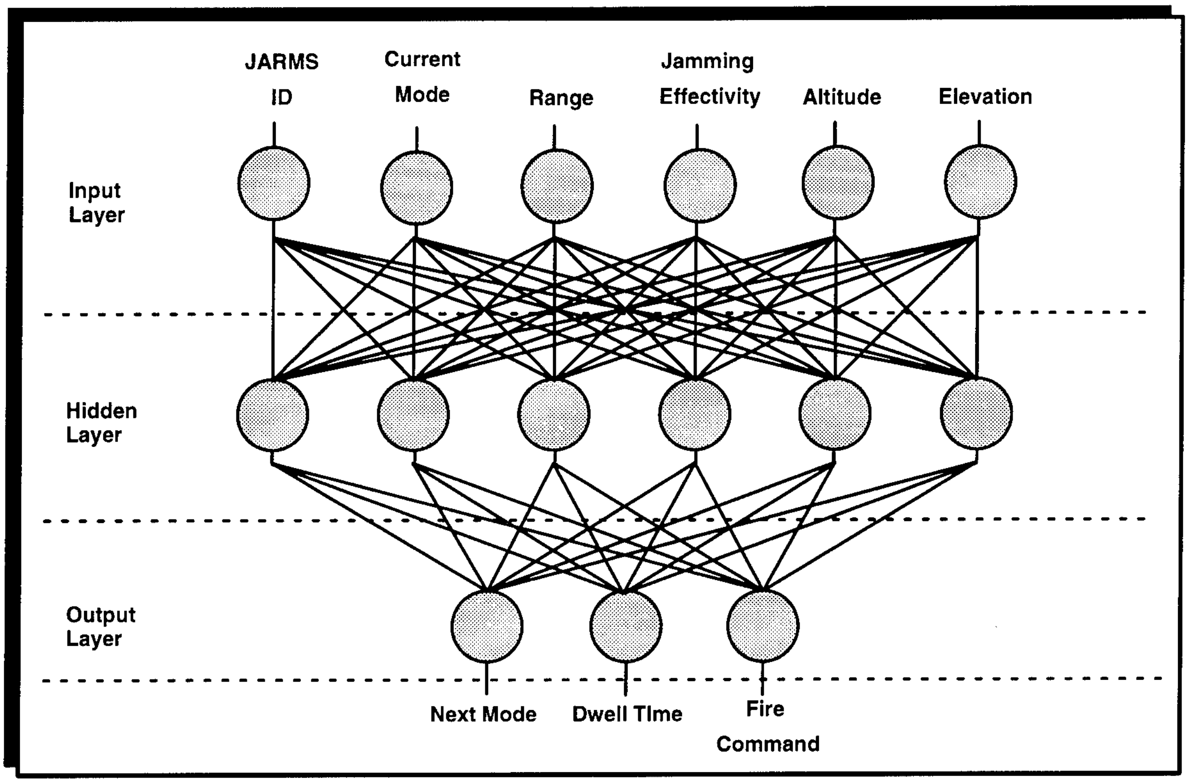


**Figure 5. SAM Type JARMS
Three Layer Neural Network Configuration**

were then tested using different data than that used for training.

For example, the network developed to model the particular TA behavior used in mode 0 was trained on inputs varying range from 40 to 160 nautical miles. It was trained to go to mode 2 for a dwell time in that mode of 30 seconds if the range is 89 nautical miles or less. If the range is 91 nautical miles or more, the next mode is 1 for a dwell time of 60 seconds in that mode.

When tested with the discrete range value of 72 nautical miles, the network yields the next mode value of 2.25 and a dwell time of 18.63 seconds. The EC system operates in distinct modes, the next mode value can be truncated to its integer value of mode 2. However, the non–integer 18.63 second dwell time can be used instead of the 20 seconds used in the existing if–then–else logic.

As can be deduced from examination, each of the SA–A TAs is not a function of all the possible inputs. Thus, a second approach was considered wherein each TA was individually modeled. This approach, like the first, yielded six networks to support modeling the SA–A. This implementation of the TAs restricted neural net inputs and outputs to only those required to capture the behavior of an individual TA. For example, inputs for the TA used in SA–A mode 3 are current mode, range, and jamming effectivity. Outputs are next mode and dwell time in the next mode.

Again, cases were developed and neural nets trained. One test case constructed for the TA used in mode 3 consists of a range input of 33 nautical miles and jamming effectivity input of 43%. For this case, the NN yielded a next mode value of 3.87 with a dwell time in that mode of 20.07 seconds. The next mode interpretation could be truncated to a mode 3 and the dwell time duration as the value concluded by the network.

The latter networks proved easier to develop than those configured in the former approach. This is due to the manner in which the problem was decomposed and limits on the number of cases that had to be learned by the networks.

CONCLUSIONS

This paper has discussed two CI methods for replacement of the current EC Simulation TA processing: Neural Networks and Fuzzy Systems. It has been shown that FSs appear to be superior to NNs for this application. The NN solution appears to be less suited for the following reasons:

- The NN solution pursued for this application is open–ended. Decomposition of the data to an optimum NN implementation was not obtained during the course of this study.
- The complexity of the problem did not allow a NN or collection of NNs to be created which produced improved performance of the TAs modelled.
- The training time of the NN was not less, and in fact much longer, than the existing method to produce new JARMS.

The FS approach appears to be stronger since TAs are more a collection of rules than a collection of data, and that for this application a knowledge based approach such as a FS is superior to the data driven solution provided by neural networks.

Utility of the FS approach was immediately obvious during this study. The FS tool permitted quick graphical construction of the SA–A TAs modelled, easy debugging, generated commented code, and produced acceptable models. However, a much greater effort than that conducted in this study will be required to produce a useful product to integrate into the existing EC Simulation.

FSs could provide an improved performance of the EC Simulation. This is due to their capability of expanding knowledge of the if–the–else logic and using computational techniques to extract better conclusions.

The FS developed is implemented in ANSI C code, which is compatible with the existing simulation. Changes would entail modifications including the new FS modules and deletion of the existing TA if–the–else structure code. The FS C source code would be cross complied on a development system and resulting object code added to the EC Simulation build.

BIBLIOGRAPHY

1. How Neural Networks Learn from Experience
   Geoffrey E. Hinton
   Scientific American, September 1992

2. Neural Networks
   Peter J. Denning
   American Scientist, September–October 1992

3. Working with Neural Nets
   Dan Hammerstrom
   IEEE Spectrum July 1993

4. Fuzzy Logic – From Concept to Implementation
   Aptronix, Inc.

OPERATIONAL PROTOTYPE FOR AN INSTRUCTOR/OPERATOR STATION

Dick Fulton and Ankur Hajare
Enhanced Technologies
P. O. Box 890886
Houston, Texas 77289-0886

Tom Diegelman
NASA Johnson Space Center

Dave Webster
CAE-Link Corporation

Abstract

This paper describes the implementation process for technology insertion into a real-time, human-in-the-loop, flight simulator of the Space Shuttle used for astronaut training. The Instructor/Operator Station (IOS) is a twelve year old, highly tailored subsystem that was not designed to easily accommodate changes in hardware and software technology. Since the Shuttle program is anticipated to run another 15 years, the objectives of the project were to identify and evaluate commercial off-the-shelf (COTS) hardware and software that meet defined requirements for the upgrade of the IOS in the training facility. The rationale for conducting the prototype was to find the best possible way to upgrade the IOS and minimize the life cycle costs for continuing operations. The paper illustrates the prototype architecture that was successfully used to establish the confidence of NASA management in the concept and to refine the technical requirements for the IOS upgrade. The paper also discusses the lessons learned in implementing an operational prototype in a complex real-time simulator as well as the plans for the future of the IOS.

Author's Biographies

Dick Fulton is the manager for systems engineering at Enhanced Technologies, Houston, Texas. Previously, he was employed by the MITRE Corporation in Houston, Texas for ten years. Dick Fulton was the subsystem manager for NASA for the IOS upgrades to the Shuttle Mission Training Facility (SMTF). Dick has a BS in Engineering and 26 years of experience in information systems. He can be reached at (713) 280-0111.

Ankur Hajare is a vice president and co-founder of Enhanced Technologies, Houston, Texas. Previously, he was employed by the MITRE Corporation and worked at the NASA Johnson Space Center for 14 years. His work in the SMTF included system design reviews, benchmarking, acquisition support, software conversion studies, computer systems performance measurement and analysis, computer systems modeling, and visual systems evaluation. Ankur has a BTech and MS in Electrical Engineering. He can be reached at (713) 280-0111 or (713) 480-4757.

Tom Diegelman is the project manager for the SMTF upgrade, and has responsible for the planning, scheduling and control of that project. Tom was responsible for the definition of the technology exploration initiatives in the SMTF upgrade. Tom has a BS in Aeronautical Engineering and an MS in Electrical Engineering. He has over 18 years experience in the simulation field in both aeronautical and petrochemical fields.

Dave Webster was the CAE-Link Project System Engineer (PSE) for IOS subsystems. He has supported the SMTF for nine years. Dave was responsible for directing and coordinating all prototyping work done by CAE-Link in the IOS area. As the PSE, Dave was responsible for all customer and associated contractor interfaces, as well as budget and scheduling activities.

OPERATIONAL PROTOTYPE FOR AN INSTRUCTOR/OPERATOR STATION

Dick Fulton, Ankur Hajare, Tom Diegelman, and Dave Webster

INTRODUCTION

The principal reasons we were successful in developing the IOS prototype is that the challenge associated with the concept of injecting new technology into the Shuttle Mission Training Facility (SMTF) was accepted as an innovative task that would become a cornerstone in a very long project.

As an agency, NASA is still quite "young" and being less than 30 years old, the JSC facilities are likewise "young." These facilities were built when the display technology was just beginning to mature. As a trainer built 15 years ago, the SMTF exhibits custom designed, specialized equipment that today would likely be purchased and installed as COTS.

Our facilities are tailored to concepts of what is both needed and feasible. Vertical integration of the facility is still evolving. This creates a large, complex set of equipment that meets the objectives of only a single program. We are faced, for the first time, with the reality of working two long term space programs concurrently. As the Space Station Freedom program comes on-line, we still must support the Shuttle program.

We need to seek out facility upgrade concepts that reflect common hardware, software, and architecture across multiple programs. The training facilities being built for the Freedom program will be mostly state-of-practice COTS products. It is all too obvious that the same COTS products could also be used in the upgrade of the SMTF and make substantial reductions in the cost of ownership for both facilities.

The IOS prototype team was required to identify the facility issues associated with technology insertion and accommodate the ongoing needs of the instructors, operators, and maintenance personnel during standalone and integrated training.

During integrated training, the Network Simulation System (NSS) provides a realistic representation of the real world Shuttle communications network. Details, such as loss-of-signal occlusion, are modeled in this simulator. The NSS is sufficiently realistic, that during integrated simulations (MCC in the loop) the ground controllers in the MCC can not tell if the Shuttle is actually flying or being simulated in the SMTF.

Instructors are required to enter reconfiguration commands in a short period of time as the simulated Shuttle communications move from one ground station to the next. As we prepared plans and requirements for the upgrade to the SMTF, the operators wanted a more efficient method of entering NSS reconfiguration commands. To really make this work, the operators proposed using workstations to provide this improved functionality.

It did not make sense to us to put in a workstation for an isolated user. We considered it essential to develop a prototype and install it in the training environment to assess the user evaluation of a new IOS subsystem. This would give us broader exposure to the new technology and provide a much more cost effective way to gather needed requirement data.

BACKGROUND

Training in the SMTF is designed around the concept of remote instruction. The instructor is not in the seat next to the trainee with direct override capability. This override capability must be simulated.

The SMTF architecture uses a central host computer system and a relatively "dumb" man-machine interface for instructors and operators. The SMTF host runs on a 40 millisecond time frame and the frames are becoming saturated. This is evident from soft transgressions. Without a complete restructure of the simulation design, it is difficult to reconfigure from flight to flight and add any nuances. This points to a need to off-load the host.

The IOS is the focal point for external interfaces in a real-time simulation. The instructors must have interactive access to the data observed in the crew station, and the operators must have the resources to control the real-time simulation loads.

All functional interfaces such as visual scenes, voice communications, flight aural cues, on-board computers, and others, must be available in the IOS.

Given the state-of-the-art in the 1970's, the requirement for even the most basic IOS required development of in-house software and hardware products. We needed to understand the existing simulation interfaces in detail. We also needed to understand the COTS products to effectively design the interface between the COTS products and existing systems.

It is important to maintain current capability while designing the in-roads of new requirements.

Little of the SMTF environment is static. Flight reconfiguration and facility modifications are needed for each flight. Many of these modifications are payload dependent.

We create a software baseline to target subsystem replacements. Baseline one is used to sustain training, and baseline two is used to implement development changes. Changes made to baseline two are handed over to operations when development is complete. In the SMTF, baselines are created at six month intervals. If we miss a baseline drop, we face a possible six month slip in our software development schedule.

We need to understand and be aware of other ongoing changes in the facility. We consider this critical to the success of a prototype. Even things we believe are cast in concrete, can be changed. We will discuss one such surprise later.

Upgrades to training capability are dependent on whether or not the existing equipment can support the change.

The IOS equipment is marginally supportable. Hardware parts are mostly available, but the systems are no longer sold as stock items. The end of the life cycle for the IOS hardware is near. Because of the custom tailored software coupling to the hardware, the software is "doomed" as well. Porting of this software is not possible. The data display systems in the IOS have been in use in excess of fifteen years. We developed the IOS page compiler in-house. It is written in assembly language on a 36 bit computer. This is a hardware and software dinosaur, and it is also not portable. This hardware and software is like a piece of flypaper, once you get hold of it very difficult to financially justify getting rid of it. We documented this in the Level A Functional Requirements document (NASA, 1991).

Page development requires simulator time. The page code is written off-line but simulator time is required to compile and test the page code and checkout the format of the page. If a page requires a change, the simulation must be brought down, the change made off-line, and then the simulation load must be rebuilt to include the page modifications. Changing pages in a training load is expensive due to the load building overhead. We rarely make small changes due to the large cost of simulator usage. Instructors are forced to work with page code that has technical problems. What should be an inexpensive task, is not, using the simulation host is driving costs up.

When the SMTF was built, we built a custom interface between the IOS and the host computer because high interrupt rates from the IOS were causing the simulation to terminate. The interrupts from the IOS had to be metered to a rate the host computer could support.

APPROACH

We wanted the new IOS to be a compatible (a plug-in component) with the both the existing and a potential COTS replacement simulation host.

We wanted to off-load the data conversion and presentation work from the simulation host computer to the IOS. This workload is very unpredictable and it makes sense to move it outside of the timing-sensitive simulation host. We decided that the MCC design, that included a Network Data Driver (NDD) with suitable "enhancement", would satisfy the IOS upgrade requirements.

The Configuration

The NDD serves as the networking "traffic cop". It intercepts, analyzes, and validates data requests from the instructor position.

The host computer data, returning to the instructor position, is converted to required format by the NDD. There are two Local Area Networks (LANs) that are controlled and monitored by the NDD. The real-time LAN distributes data to the instructor positions. The general purpose LAN is used for full dialog between the NDD and the instructor position.

The hardware interface between the NDD and the simulation host is COTS, used by many vendors.

The software interface is through the symbol dictionary The symbol dictionary provides the memory address translation between the requested data term and the virtual memory of the simulation host. This feature provides us the ability to build displays outside of the simulation host, makes the IOS upgrade concept possible. Regardless of the re-host approach, there will always be some form of change of the data output by the simulation host. Any changes to the NDD is minimal and localized to the address resolution software or the data conversion routines.

The Team

We had a good combination of talent and dedication.

The prototype team consisted of five engineers. The Project System Engineer was responsible for budget, schedule, and customer interface. One System Engineer was responsible for the engineering of the technical aspect of the task and specifically the NDD.

One Engineer was assigned to the workstation. He was responsible for the code and pages running on the workstation. This engineer was also responsible for all the software administrative functions. One hardware engineer was assigned to the hardware interface to the simulation host. Another engineer was responsible for "hooks" into the simulation code for data gathering

The Users

Because few upgrades in technology have been implemented in this time frame, the impact of technology insertion is very significant to the users.

The IOS upgrade is a unique departure from the past. It is mandatory to implement changes in such a way that the user can be brought "on board" in small steps.

We wanted the users to help create the design. This will make the end product more acceptable. We asked the users to play a critical role in the requirements definition and design phases. Users were brought in weekly to work with the prototype. Great care and attention were given to the user inputs, making the user an integral part of the process.

We initially emphasized replicating the capabilities of the IOS as the instructors see them today. The users identified a core set of displays needed to perform a majority of their routine activities. This core set was developed in a "look-a-like" fashion.

The users are given small prescribed doses of change to keep familiarity alive and not perturb every day astronaut training.

We introduced new twists about every three months. For example, video that requires its own monitor was routed through a "window" to the new display hardware. This was not a particularly large change, because the scenes had not been modified, but great care was given to make sure that the new IOS capability would be compatible with the visual upgrade going on in parallel.

RESULTS

Preparation

Our first, and perhaps most difficult, task was to obtain management approval for the prototype. This process involves preparing a design package that includes the following significant parts:

- Required floor plan modifications
- Electric power needs
- Temperature control, chilled water and air
- Data path identification
- Security and equipment isolation

We worked on the approval process for two years. During this time we made one major change to the design package. This change added a second training base where the prototype would be installed.

Also during the approval process, we proceeded to establish the initial prototype in the Link lab. We built a stripped down real-time load and took crude timing measurements.

Next, we had to convince the SMTF sustaining contractor that what we were adding to the simulator host computers was a small and manageable risk. The SMTF has two temperamental processes.

- All tasks must be completed within 40 ms
- I/O Channel use has little to spare

We worked with the operations and sustaining engineering people to gain their confidence in our plan.

Installation

We go through a process that assures we cover all important aspects of change to the SMTF. In some cases we need the specific technical details for change from other organizations, and in other cases we have to supply the technical detail ourselves. This process may be different for you because of the organization of your support services.

Installation of the prototype was done in three stages. First, we needed a development facility where initial work could be done without the risks of interrupting astronaut training. The first prototype was installed in the development contractors facility. We were able to build a realistic capability for demonstration, but we could not tie into the real time data available from the simulation host computers. We used this capability to increase user, management, and sustaining contractor confidence in the prototype. This installation also provided an opportunity for the development contractor to learn a different COTS programming language.

Second, we installed the prototype in a training base that doubles as a development facility. This installation required close coordination with other SMTF upgrade activities that were being done at the same time. This training base is being upgraded to full training capability. At one point, we were planning a demonstration that conflicted with dismantling activities on that training base.

This second installation provided the first opportunity to bind the prototype to the real-time data flow without the risk of interrupting training. At this point, we could and did demonstrate the enhanced capabilities proposed for the IOS upgrade.

Third, we installed the prototype in an active training base. This installation gives the instructors an opportunity use the prototype during a real training session. It also provides a way to get user feedback. This feedback closes the loop on integrating the user in the IOS upgrade development process.

The third installation came with several caveats. We were not to cause the simulation to terminate abnormally. (Murphy's Law "If it can happen, it will happen" got to us - we did it!). We were not to run the prototype during an integrated simulation involving the MCC and thereby risk a much more costly simulation termination (Murphy got to us again!).

We spent several weeks trying to discover what went wrong that caused the simulation to terminate. Early in the investigation, we discovered that the data buffers were corrupted. Thus, we found out that some other upgrade task broke the concrete!

We set our buffer switching algorithm based on the 25 hertz frame dispatching rate. Due to another problem in the simulator, the first frame at simulator initialization was skipped. This caused our buffer switching algorithm to use the same buffer for both input and output. We had to wait for the next six month baseline delivery to implement a correction for this buffer switching problem.

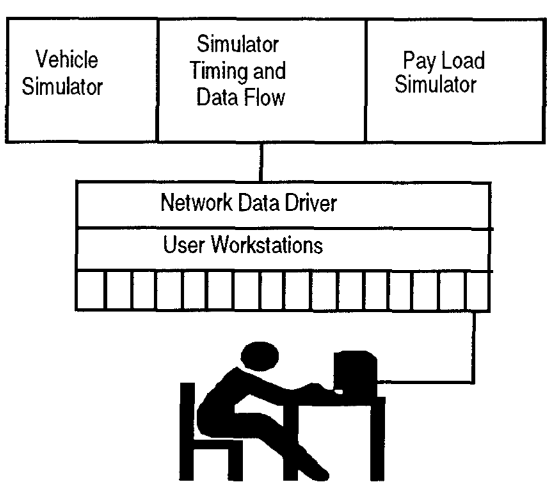


Figure 1
Generic IOS Configuration

CONCLUSIONS

We determined that the MCC network data driver and workstation configuration can be used in the SMTF. The generic configuration is shown in figure 1.

You should note that the implementation of this configuration is highly dependent on the geography of the training facility. In the closest of spaces, you may be able to package the simulator host computer, the network data driver and user workstations in the same 19 inch rack.

Metrics

We accomplished the goal of identifying requirements for the next generation IOS subsystem.

- The total simulation host computer load consists of about 600,000 lines of code. About 200,000 lines of code are executed every 40 milliseconds. The prototype added 5,000 lines of code to the 40 millisecond work load.

- The data path between the simulation host computer and the network data driver must be able to sustain a synchronous data path capable of transferring 1 megabytes per second.

- The network data driver must execute 2,500 lines of code every 40 milliseconds.

- The network data driver must be able to sustain a synchronous data path capable of transferring 1 megabyte per second.

- The network data driver must be able to handle a peak asynchronous data rate of 25 messages per second with a maximum of 8,000 bytes per message.

- The IOS must be capable of continuously logging data at rate of 8,000 bytes per second for each of a maximum of twenty five users. The NDD must be able to log time stamped events up to a maximum of 125 events per user per second.

- The IOS must be able to execute 12,000 lines of source code per user per second.

- The IOS must be able to display in color a maximum of 12 windows (2,000 single precision data terms) including one video scene per user per second. All data shown on user displays must be time homogeneous, this means text and graphics.

The network data driver must be benchmarked at 40 "SPECMARKS". Each user must also have processing capability bench marked at a bare minimum of 25 "SPECMARKS". We recognize that the user processing needs will increase with new training requirements and with upgrades in COTS software products. We believe it is desirable to find a computer vendor that has a linearly scaleable product. Fortunately,, there are several vendors that offer single board computers that can be upgraded with new technology at the board level.

Lessons Learned

We had one situation we could have handled better. At the time we were seeking approval for the IOS prototype, we were also participating in another prototype with a more limited scope but addressing the same issue - "Improving the User Interface to the SMTF". The other prototype was targeted specifically at display design using COTS graphical products. This prototype had user support, money, equipment, and a small but talented staff.

A significant difference between the two prototypes was the use of real time data to drive the displays. The other prototype was using mock-up displays that did not react to the dynamics of any real time data flow. We attempted to merge the two prototypes into a single effort, but we were unsuccessful. The other prototype was dissolved, the staff was reassigned, the money was withdrawn, and the equipment was reallocated to other needs. What remained was the user support and the documentation of their accomplishments to date.

At issue was the type of funding between the two prototypes. Funding for the IOS prototype was development, and funding for the other prototype was research. Also at issue was our need to scope the amount of effort we needed to build user displays. In retrospect, we should have scoped the full effort for building user displays and taken advantage of anything the other prototype could have provided. By drawing attention to the similarities and differences of the two prototypes, we lost the support and possible contributions the other prototype could have provided.

We demonstrated that, even in this completely "in-house" software environment like in the SMTF, COTS products could be introduced with minimal difficulty and yield superior results. This is one of the most significant effects of the IOS prototype.

The IOS prototype was developed in a very open environment. Periodic demonstrations, working group meetings with the users, and hands-on user involvement were intricate parts of the IOS development. The demonstrations were scheduled on the basis of capability deliveries. Each delivery was reviewed by the users, written comments were taken and the users were able to mold the product through the development cycle. This development method keeps the user community involved.

Recommendations

We did some things correctly and we want to share these with you.

Prepare an implementation plan that shows how the prototype will evolve. There are three important points that attract attention. First, be specific about the expected results and deliverables of the prototype. Second, the plan should include several demonstration sessions so management will recognize significant progress when reported. Third, show that the prototype has a definite development termination date, even if it is planned to be left in an operational state.

Select the prototype team carefully. The lead team members must be convinced the effort will produce worthy results, that the approach is technically sound and attainable goals set within the plan schedule. There is a natural conflict of interest between the selection of prototype personnel and the determination of the prototype schedule. In order to shorten the prototype schedule and produce meaningful results, you will select your best performers to work on the prototype. On the other hand when problems arise in profit centers, these same people will be pulled away from the prototype to solve these problems.

A successful prototype does not come without commitment. Therefore, our recommendation is to involve the technical lead personnel in the preparation of the prototype implementation plan. Listen to them; after all, their careers and credibility are also at stake.

Avoid committing to a comprehensive set of documentation requirements. The prototype plan should be synchronized with the development schedule. Deliverables from the prototype should meet the needs to document the requirements of the development project. Let the development project bear the cost and schedule impact for comprehensive documentation. Documentation needs will refine themselves as the prototype effort evolves.

Be prepared to defend the prototype at every status meeting. Your efforts can be attacked from the most unsuspecting corners of your environment at any time. There is an ancient saying "Under the tree with the ripest apples, lie the biggest clubs." If the prototype is not attacked, it is probably seen as a failure in the making.

The Future

The vision for the future of the SMTF complex with respect to the IOS has been historically difficult to formulate. However, recently NASA has made agency wide and center wide attempts to establish guidelines, much analogous to the "why and how to prototype" guidelines the authors established during the IOS prototype effort. There is clearly a charter to infuse technology, but in a manner consistent with information system technology in NASA facilities. The priorities defined are to "restore, maintain and construct, in that order" information systems. Clearly, innovation is required in order to attain the goal of a sustained 15% Shuttle Program cost reduction by 1996. This translates into improved facility management philosophy, especially with respect to information processing centers

To tie dissimilar facilities and systems together is most easily accomplished under the auspices of common procurements for equipment, allowing divergent applications and operation environments to function with common hardware and software, most of which are commercial-off-the-shelf (COTS).

Against this vision, the "futures" of the IOS developed. A brief discussion of the major initiatives identified and considered to be candidates for follow-on system engineering studies are discussed.

With the inclusion of high powered workstations into the IOS platform, the ability to utilize LAN connectivity introduces the separability of the individual workstation to perform dedicated tasks. The merging of the single system trainer functionality (SST), currently a host / dumb terminal based system, into the full fidelity SMTF is seen as very attractive. The necessary models to create the SST environment would require dedicated MIPS, models, sequencers, user interface, and a rather low-fidelity functional flight software model as a "driver" for the entire system . In place of the "hard" configuration of switches and panels, the workstation's multiple graphics heads form the "panels".

The entire system is then linked together by the predetermined sequencer (i.e., the requirements for what system is being trained) and is executed stand-alone from the full SMTF base. If multiple students required multiple stations, multiple workstations would be networked together. Again, this would be a predefined sequencer that "built" the environment. Upon completion of the session, the workstations are reconnected to the LAN (figure 2).

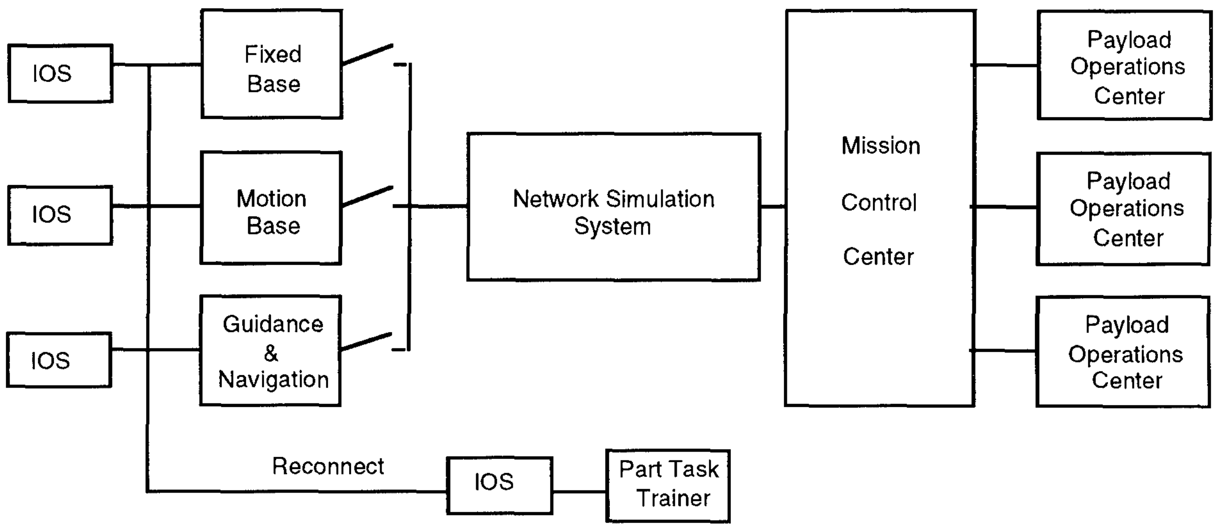


Figure 2
Facility Level Configuration

REFERENCE

NASA Johnson Space Center (1991). Level A Functional Requirements for the Shuttle Mission Training Facility Equipment Replacement or Upgrade Step 4, (Draft). Houston, TX.

Alphabetical Listing of Authors

Alphabetical Listing of Paper Titles